ระบบแปลภาษามือด้วยเซนเซอร์สามมิติ Leap Motion

The ASL translator system by Leap Motion 3D sensors

ศุภวัฒน์ คำขุ้ย
SUPPAWAT KHAMKUI

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2557

ระบบแปลภาษามือด้วยเซนเซอร์สามมิติ Leap Motion

The ASL translator system by Leap Motion 3D sensors

โดย

ศุภวัฒน์ คำขุ้ย

อาจารย์ที่ปรึกษา

ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2557

ปริญญานิพนธ์ปีการศึกษา 2557

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแปลภาษามือด้วยเซนเซอร์สามมิติ Leap Motion

The ASL translator system by Leap Motion 3D sensors

ผู้จัดทำ นายศุภวัฒน์ คำขุ้ย รหัสประจำตัว 54011295

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

๐๗ ส.ค.

(ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ)

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์	ระบบแปลภาษามือด้วยเซนเซอร์สามมิติ Leap Motion
นักศึกษา	นายศุภวัฒน์ คำขุ้ย รหัสประจำตัว 54011295
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2557
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ

บทคัดย่อ

ปริญญานิพนธ์นี้พัฒนาระบบแปลภาษามือเป็นตัวอักษร โดยใช้การตรวจจับนิ้วมือทั้ง 5 นิ้ว แบบไม่สัมผัสในรูปแบบ 3 มิติ โดยเครื่องมือที่ใช้ในการพัฒนาโครงงานได้แก่ อุปกรณ์เซนเซอร์สามมิติ Leap motion เชื่อมต่อกับคอมพิวเตอร์ และพัฒนาโดยโปรแกรมด้วย Visual Studio 2010 ทำการตรวจจับนิ้วมือที่แสดงท่าทางของตัวอักษรภาษาอังกฤษจำนวน 26 ตัวอักษร ทำการเก็บผลทางสถิติเป็นรายตัวอักษรเพื่อหาค่าที่แม่นยำ เพื่อให้ได้ระบบที่สามารถช่วยแปลภาษามือ และ เพื่ออำนวยความสะดวก ในการสื่อสารกับผู้พิการทางการได้ยิน

Thesis Title	The ASL translator system by Leap Motion 3D sensors
Student	Mr.Suppawat Khamkui Student ID 54011295
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2014
Thesis Advisor	Asst.Prof.Dr.Pattarapong Phasukkit

ABSTRACT

The purpose of this thesis were developed sign language to alphabets system used by 3D unobtrusive detect with 5 fingers. We used Leap motion (3D sensors) connected with computer and developed by Visual Studio 2010. Detect fingers that show gesture of 24 English alphabets. The statistics result of each alphabets affect to accurate data to use for translate sign language device and convenience to communicate with deafened people.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงลงได้ด้วยความกรุณาของ ผศ.ดร.ภัทรพงษ์ ผาสุขกิจ อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษาและติดตามการดำเนินการอย่างใกล้ชิด ชี้แนะแนวทางวิธีการ ซึ่งเป็นประโยชน์ต่อการดำเนินการวิจัย ข้าพเจ้าขอกราบขอบคุณเป็นอย่างสูง

ขอบคุณรุ่นพี่ และเพื่อนๆ ที่คอยให้กำลังใจและความช่วยเหลือตลอดจนให้คำปรึกษาแก่ข้าพเจ้าในการทำปริญญานิพนธ์เล่มนี้ให้เสร็จสมบูรณ์

สุดท้ายนี้ขอกราบเท้าขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำเนิด เลี้ยงดู อบรมและสนับสนุนการศึกษาเป็นอย่างดี

ศุภวัฒน์ คำขุ้ย

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในสังคมไทยมีผู้บุคคลที่มีความบกพร่องทางการได้ยิน เป็นจำนวนไม่น้อย ซึ่งบุคคลเหล่านี้มีปัญหาทางการสื่อสารด้วยการพูดคุย จึงจำเป็นต้องใช้ภาษามือในการสื่อสารในชีวิตประจำวัน เพื่อที่จะให้บุคคลทั่วไปสามารถสื่อสารกับบุคคลที่มีความบกพร่องทางการได้ยิน จึงมีแนวคิดที่จะสร้างระบบการแปลภาษามือเป็นตัวอักษรภาษาอังกฤษ เพื่อที่จะทำให้สะดวกต่อการสื่อสาร ระหว่างบุคคลที่มีความบกพร่องทางการได้ยิน และบุคคลทั่วไป

1.2 วัตถุประสงค์ของการวิจัย

1) เพื่อศึกษาและพัฒนาการแปลภาษามือเป็นตัวอักษรภาษาอังกฤษ โดยการใช้กล้องตรวจจับข้อมือและมือ ในระบบสามมิติ

2) เพื่อช่วยเหลือบุคคลที่มีความบกพร่องหรือพิการทางการได้ยิน ช่วยเพิ่มประสิทธิภาพในการสื่อสารกับบุคคลทั่วไปได้สะดวกและง่ายยิ่งขึ้น

1.3 สมมติฐานของการวิจัย

Leap Motion จะทำการจับการเคลื่อนไหวของมือ และทำการวิเคราะห์ลักษณะท่าทาง เมื่อตรงตามเงื่อนไข ก็จะสามารถแปลเป็นตัวอักษรภาษาอังกฤษจากภาษามือได้

1.4 ทฤษฎีหรือแนวคิดการวิจัย

แนวคิดมาจากบุคคลที่มีความบกพร่องทางการได้ยิน บุคคลที่มีความบกพร่องทางการได้ยิน หมายถึง ผู้ที่สูญเสียสมรรถภาพทางการได้ยิน เนื่องจากอวัยวะการได้ยินเช่น ประสารทหูเสื่อมหรือพิการทำให้ไม่ได้ยินเสียงต่างๆ หรือได้ยินไม่ชัดและสูญเสียการได้ยินระหว่าง 26 – 90 เดซิเบล ทำให้มีผลกระทบต่อบุคคลนั้นๆ

ลักษณะของความบกพร่องทางการได้ยิน สามารถแบ่งเป็น 2 ประเภทได้แก่

- หูตึง (Hard of Hearing) หมายถึง การที่บุคคลมีข้อกำจัดในการปฏิบัติกิจกรรมใผชีวิตประจำวัน หรือ การเข้าไปมีส่วนร่วมในกิจกรรมทางสังคม เป็นผลมาจากการมีความบกพร่องในการได้ยินจนไม่สามารถรับข้อมูลผ่านทางการได้ยินเมื่อทำการวัดการได้ยินที่ความถี่ 500 , 1,000 และ 2,000 เฮริตซ์ ในหูข้างที่ได้ยินดีกว่าจะสูญเสียการได้ยินที่ความดังของเสียงตั้งแต่ 40 จนถึง 90 เดซิเบล คนหูตึง หมายถึง บุคคลที่มีการได้ยินเหลืออยู่เพียงพอที่จะได้ยินการพูดผ่านทางการได้ยิน โดยทั่วไปจะใส่เครื่องช่วยฟัง ซึ่งหากตรวจวัดการได้ยินจะมีสูญเสียการได้ยินน้อยกว่า 90 เดซิเบล ลงมาถึง 26 เดซิเบล

- หูหนวก (Deafness) หมายถึง การที่บุคคลมีข้อกำจัดในการปฏิบัติกิจกรรมในชีวิตประจำวัน หรือ การเข้าไปมีส่วนร่วมในกิจกรรมทางสังคม เป็นผลมาจากการมีความบกพร่องในการได้ยินจนไม่สามารถรับข้อมูลผ่านทางการได้ยินเมื่อทำการวัดการได้ยินที่ความถี่ 500 , 1,000 และ 2,000 เฮริตซ์ ในหูข้างที่ได้ยินดีกว่าจะสูญเสียการได้ยินที่ความดังของเสียง ตั้งแต่ 90 เดซิเบล ขึ้นไป คนหูหนวก หมายถึง บุคคลที่สูญเสียการได้ยินมากจนไม่สามารถเข้าใจการพูดผ่านทางการได้ยินไม่ว่าจะใส่หรือไม่ใส่เครื่องช่วยฟัง ซึ่งหากการตรวจวัดการได้ยินจะมีการสูญเสียการได้ยินน้อยกว่า 90 เดซิเบล ลงมาถึง 26 เดซิเบล

บุคคลที่มีความบกพร่องทางการได้ยินทั้ง 2 แบบจึงจำเป็นต้องใช้ภาษามือในการดำเนินชีวิต ภาษามือจะมีลักษณะท่าทางที่แตกต่างกันไปตามตัวอักษร ทำให้เกิดแนวคิดที่จะทำระบบการแปลภาษามือเป็นตัวอักษรสำหรับบุคคลที่มีความบกพร่องทางการได้ยิน เพื่อใช้สื่อสารกับบุคคลทั่วไป

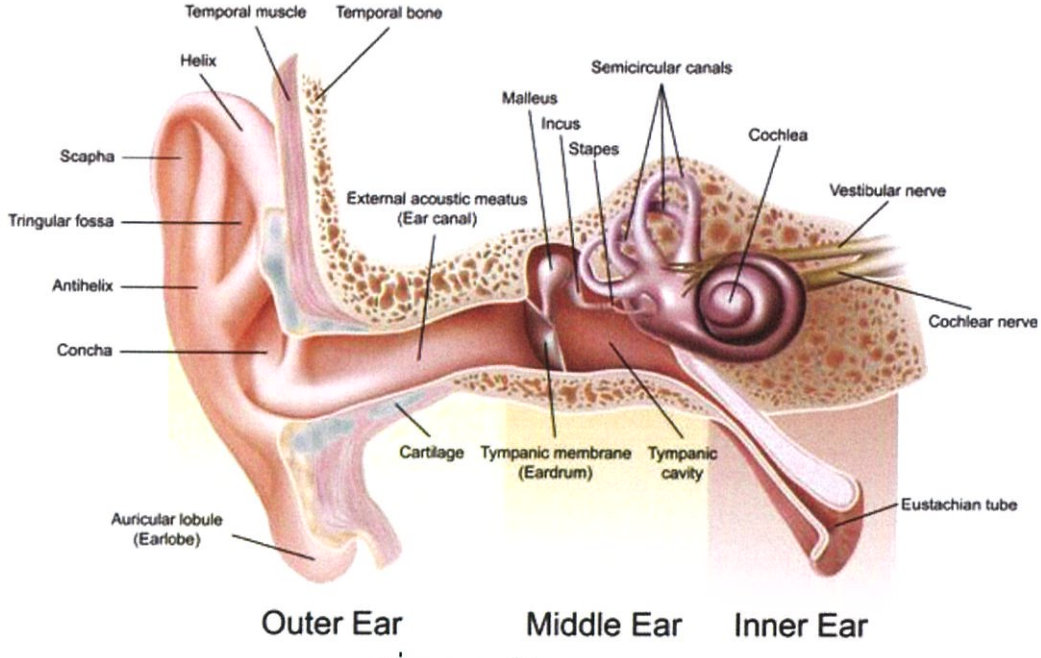
1.5 ขอบเขตของการวิจัย

1) เพื่อศึกษาการแปลภาษามือเป็นตัวอักษรภาษาอังกฤษได้
2) สามารถแปลภาษามือเป็นตัวอักษรภาษาอังกฤษได้อย่างแม่นยำ
3) สามารถแปลเป็นตัวอักษรภาษาอังกฤษได้ทุกตัวอักษร

บทที่ 2
หลักการและทฤษฎี

2.1 หูและการได้ยิน

หู คืออวัยวะรับรู้ของร่างกาย ซึ่งหน้าที่ของหูมีหน้าที่เปลี่ยนการสั่นสะเทือนทางกายภาพ (Physical Vibration) ไปเป็นสัญญาณประสาท (Nerve Impulse) โดยมีองค์ประกอบ 3 ส่วนหลักๆ ได้แก่ หูชั้นนอก หูชั้นกลางและหูชั้นใน จากข้อมูลของ Audiology specialists LLC ดังในรูปที่ 2.1



รูปที่ 2.1 องค์ประกอบของหู

ที่มา : http://www.audiologyspecialists.com/UploadedFiles/Images/anatomy-of-ear.jpg/

หูชั้นนอก ประกอบด้วย ใบหู (Pinna) และช่องหู (Ear Canal) โดยมีหน้าที่หลักคือ การรับเสียงหรือทางผ่านเสียงเพื่อเข้าสู่หูในส่วนถัดไป คือ หูชั้นกลาง ประกอบด้วย แก้วหูหรือเยื่อแก้วหู (Eardrum หรือ Tympanic Membrane) ซึ่งมีหน้าที่แปลงเสียงเป็นแรงสั่นสะเทือนไปยังกระดูก 3 ส่วน ในหูชั้นกลาง ได้แก่ กระดูกค้อน (Malleus) ทั่ง (Incus) และโกลน (Stapes) โดยกระดูกชิ้นเล็กๆ ทั้ง 3 สามชิ้นนี้ ทำหน้าที่ส่งการสั่นสะเทือนไปยังหูชั้นในต่อไป หูชั้นในหรือโคเคลีย (Cochlea) ประกอบด้วยน้ำและเซลล์ขน (Hair Cell) ซึ่งมีการรับและการตอบสนองด้วยความไวสูง โครงสร้างเล็กๆนี้ จะเคลื่อนไหวเมื่อถูกกระตุ้นด้วยความสั่นสะเทือนของเสียงและส่งไปยังเส้นประสาทรับเสียง (Auditory Nerve) ซึ่งสมองจะมีหน้าที่รับสัญญาณประสาทนี้เข้าไปเพื่อแปลความหมายต่อไป นอกจากหูชั้นในจะมีหน้าที่รับรู้เสียงแล้ว ยังมีหน้าที่ในการรักษาสมดุลร่างกายได้ด้วยเช่นเดียวกัน[8]

ปกติเสียงของมนุษย์สามารถรับรู้จะมีความถี่อยู่ในช่วง 20 – 20,000 เฮิร์ต (Hertz) ซึ่งความถี่เสียงที่พบได้ปกติทั่วไปในชีวิตประจำวันอยู่ในช่วง 125 – 8,000 เฮิร์ต แต่ช่วงความถี่ของการพูดคุยสื่อสารอยู่ในช่วงความถี่ระหว่าง 500 – 2,000 เฮิร์ต เท่านั้น โดยความถี่ที่หูจะตอบสนองได้ดีที่สุดอยู่ในช่วง 3,000 – 4,000 เฮิร์ต เนื่องจากมี Amplify Mechanism มากที่สุดภายในช่องหู ดังนั้นเสียงกระตุ้นที่อยู่ในช่วงความถี่ดังกล่าวมีผลทำให้ hair cell ที่ทำงานในช่วงความถี่นี้ถูกทำลายได้มากที่สุด จึงอธิบายได้ว่าเหตุใดการสูญเสียการได้ยินที่เกิดจากสัญญาณรบกวนจึงเริ่มเสียหายที่ความถี่ 3,000 – 4,000 เฮิร์ต[3]

2.2 บุคคลที่มีความบกพร่องทางการได้ยิน

ผู้ที่สูญเสียสมรรถภาพทางการได้ยิน เนื่องจากอวัยวะการได้ยินเช่น ประสารทหูเสื่อมหรือพิการทำให้ไม่ได้ยินเสียงต่างๆ หรือได้ยินไม่ชัดและสูญเสียการได้ยินระหว่าง 26 – 90 เดซิเบล ทำให้มีผลกระทบต่อบุคคลนั้นๆ

2.2.1 ลักษณะของความบกพร่องทางการได้ยิน

สามารถแบ่งเป็น 2 ประเภทได้แก่

2.2.1.1 หูตึง (Hard of Hearing)

การที่บุคคลมีข้อกำจัดในการปฏิบัติกิจกรรมในชีวิตประจำวัน หรือ การเข้าไปมีส่วนร่วมในกิจกรรมทางสังคมเป็นผลมาจากการมีความบกพร่องในการได้ยินจนไม่สามารถรับข้อมูลผ่านทางการได้ยินเมื่อทำการวัดการได้ยินที่ความถี่ 500 , 1,000 และ 2,000 เฮริตซ์ ในหูข้างที่ได้ยินดีกว่าจะสูญเสียการได้ยินที่ความดังของเสียงตั้งแต่ 40 จนถึง 90 เดซิเบล คนหูตึง หมายถึง บุคคลที่มีการได้ยินเหลืออยู่เพียงพอที่จะได้ยินการพูดผ่านทางการได้ยินโดยทั่วไปจะใส่เครื่องช่วยฟัง ซึ่งหากตรวจวัดการได้ยินจะมีสูญเสียการได้ยินน้อยกว่า 90 เดซิเบล ลงมาถึง 26 เดซิเบล

2.2.1.2 หูหนวก (Deafness)

การที่บุคคลมีข้อกำจัดในการปฏิบัติกิจกรรมในชีวิตประจำวัน หรือ การเข้าไปมีส่วนร่วมในกิจกรรมทางสังคม เป็นผลมาจากการมีความบกพร่องในการได้ยินจนไม่สามารถรับข้อมูลผ่านทางการได้ยินเมื่อทำการวัดการได้ยินที่ความถี่ 500 , 1,000 และ 2,000 เฮริตซ์ ในหูข้างที่ได้ยินดีกว่าจะสูญเสียการได้ยินที่ความดังของเสียง ตั้งแต่ 90 เดซิเบล ขึ้นไป คนหูหนวก หมายถึง บุคคลที่สูญเสียการได้ยินมากจนไม่สามารถเข้าใจการพูดผ่านทางการได้ยินไม่ว่าจะใส่หรือไม่ใส่เครื่องช่วยฟัง ซึ่งหากการตรวจวัดการได้ยินจะมีการสูญเสียการได้ยินน้อยกว่า 90 เดซิเบล ลงมาถึง 26 เดซิเบล

บุคคลที่มีความบกพร่องทางการได้ยินทั้ง 2 แบบจึงจำเป็นต้องใช้ภาษามือในการดำเนินชีวิต ภาษามือจะมีลักษณะท่าทางที่แตกต่างกันไปตามตัวอักษร ทำให้เกิดแนวคิดที่จะทำระบบการแปลภาษามือเป็นตัวอักษรสำหรับบุคคลที่มีความบกพร่องทางการได้ยิน เพื่อใช้สื่อสารกับบุคคลทั่วไป

2.2.2 ลักษณะของบุคคลที่มีความบกพร่องทางการได้ยิน

โดยทั่วไปสามารถสังเกตได้ในลักษณะต่างๆ ดังนี้

2.2.2.1 บุคคลที่มีความบกพร่องทางการได้ยินอาจจะมีปัญหาในการพูด

พูดได้น้อย หรือ พูดไม่ชัด ต้องใช้ภาษาท่าทางหรือภาษามือ ทั้งนี้ขึ้นอยู่กับระยะเวลาที่สูญเสียการได้ยิน เช่น หากบุคคลสูญเสียการได้ยินตั้งแต่กำเนิดอาจจะมีปัญหาในการพูดมากกว่าบุคคลที่สูญเสียการได้ยินหลังจากภาษาพูดแล้ว และการที่ได้รับบุคคลได้รับการสอนพูดตั้งแต่เด็กหรือไม่

2.2.2.2 บุคคลที่มีความบกพร่องทางการได้ยินอาจจะมีปัญหาในการใช้ภาษา

ใช้ภาษาค่อนข้างจำกัดและไม่ถูกหลักไวยากรณ์ รู้คำศัพท์ต่างๆในวงจำกัด ไม่สามารถนำคำศัพท์ไปใช้ในประโยคที่หลากหลายหรือต่างสถานการณ์ อาจจะมีคำสลับที่คำ หรือข้อความในประโยคที่ส่งผลให้การเขียนบกพร่องด้วย เช่น “เหตุหมาย” หรือ “กระเพาะ” เป็น “กระเฉพาะ”

- ผลสัมฤทธิ์ทางการเรียน บุคคลที่มีความบกพร่องทางการได้ยิน ส่วนใหญ่จะมีผลสัมฤทธิ์ทางการเรียนต่ำ ทั้งนี้เนื่องจากความบกพร่องในการฟังทำให้มีทักษะทางภาษาจำกัด และวิธีการสอน และการวัดผลของครูที่ไม่เหมาะสม

- การปรับตัว บุคคลที่มีความบกพร่องทางการได้ยินมีปัญหาในการปรับตัวเนื่องจากไม่สามารถสื่อสาร กับผู้อื่นให้เข้าใจ เกิดความขับข้องใจและส่งผลต่อพฤติกรรมที่แสดงออกมา

- ลักษณะอื่นๆ ได้แก่

  - เวลาฟังมักจะจ้องมองปากหรือจ้องหน้าผู้พูด
  - เสียงพูดแปลก อาจเป็นเสียงต่ำ หรือแหบผิดปกติ
  - ไม่สามารถปฏิบัติตามคำสั่งได้
  - ไม่มีปฏิกิริยาต่อเสียงที่ดังๆ เช่น เสียงแตรรถยนต์ เป็นต้น

2.2.3 วิธีสอนเฉพาะสำหรับเด็กที่มีความบกพร่องทางการได้ยิน

2.2.3.1 การสอนการใช้ภาษามือ

เด็กที่มีความบกพร่องทางการได้ยินมาสามารถที่จะเข้าใจและใช้ภาษาพูดได้ทั้งระบบ จึงมีความจำเป็นต้องใช้ภาษามือ และการสะกดตัวอักษรด้วยนิ้วมือ ช่วยเสริมความเข้าใจในภาษาพูดและเขียน โดยผู้สอนจะต้องทำหน้าที่เป็นล่ามภาษามือเพื่อเสริมความเข้าใจ และการพูดบรรยายความรู้ทางด้านวิชาการก็ต้องใช้การสะกดตัวอักษรด้วยนิ้วมือ เพื่อประโยชน์ต่อการเขียน และบันทึกเป็นภาษาเขียนให้สมบูรณ์ถูกต้องตรงกับความจริง เนื่องจากภาษามือคือภาษาสำหรับคนที่มีความบกพร่องทางการได้ยิน เป็นภาษาที่แสดงออกด้วยการใช้มือในท่าทางต่างๆประกอบกับสีหน้าและกิริยาท่าทางประกอบในการสื่อความหมาย และ ถ่ายทอดอารมณ์แทนการพูด โดยจะมีความแตกต่างกันตามขนบธรรมเนียมประเพณี วัฒนธรรมแต่ละถิ่นเช่นเดียวกับภาษาพูด

2.2.3.2 การสอนฟังและพูด

เนื่องจากการฟังและการพูด เป็นส่วนหนึ่งของการติดต่อสื่อสารที่จะช่วยให้ทุกคนรู้และเข้าใจสิ่งต่างๆ สำหรับเด็กที่มีความยกพร่องทางการได้ยิน เนื่องจากมีความบกพร่องทางการได้ยิน จึงเป็นสาเหตุให้เด็กที่มีความบกพร่องทางการได้ยิน ไม่สามารถเรียนรู้การพูดและการใช้ภาษาพูดได้ดี ดังนั้นจึงมีความจำเป็นต้องฝึกพูดให้เด็กที่มีความบกพร่องทางการได้ยิน เพื่อจะได้ใช้การได้ยินที่เหลืออยู่ให้เป็นประโยชน์มากที่สุดในการเข้าใจภาษาพูด และมีความสามารถใช้ภาษาพูดในการสื่อความหมายกับบุคคลอื่นในสังคมได้

2.2.3.3 การสอนอ่านริมฝีปาก

เป็นวิธีที่เด็กที่มีความบกพร่องทางการได้ยินในการรับภาษาพูดจากผู้อื่นและเป็นสิ่งแรกที่เด็กที่มีความบกพร่องทางการได้ยินจะต้องเรียนรู้วิธีการอ่านริมฝีปากตั้งแต่ครั้งแรกที่เรียนภาษา เพราะเป็นสิ่งที่เด็กจำเป็นต้องใช้ไปตลอดชีวิต ผู้ฝึกควรให้เด็กได้ศึกษาวิธีการอ่านริมฝีปากเพื่อการนำไปใช้ชีวิตประจำวัน และต้องมีความเข้าใจว่า การอ่านริมฝีปาก คือ ศิลปะของการเข้าใจภาษาพูดโดยการแปลความหมายจากการสังเกตการเคลื่อนไหวของอวัยวะในการพูด เช่น ใบหน้า ลิ้น ขากรรไกร และ คอ[4]

2.3 ภาษามือ

จากการศึกษาของ พญ.เพ็ญมาศ ธรรมศรัณยู, 2554[2] ภาษาสำหรับคนหูหนวก ใช้มือ สีหน้าและกิริยาท่าทาง ประกอบในการสื่อความหมาย และถ่ายทอดอารมณ์แทนการพูด ภาษามือของแต่ละชาติมีความแตกต่างกันเช่นเดียวกับภาษาพูด ซึ่งแตกต่างกันตามขนบธรรมเนียม ประเพณี วัฒนธรรม และลักษณะภูมิศาสตร์ เช่น ภาษามือจีน ภาษามืออเมริกัน และภาษามือไทย เป็นต้น ภาษามือเป็นภาษาที่นักการศึกษาทางด้านการศึกษาของคนหูหนวกตกลงและยอมรับกันแล้วว่า เป็นภาษาหนึ่งสำหรับติดต่อ สื่อความหมาย ระหว่างคนหูหนวกกับคนหูหนวกด้วยกัน และระหว่างคนหูหนวกกับคนหูดี ภาษาอังกฤษเรียกการสื่อสารด้วยมือนี้ว่า Sign Language หรือ Manual Communication[9]

2.3.1 ภาษามือแบบอเมริกัน (ASL)

ชื่อเต็มคือ American Sign Language เป็นภาษาที่บุคคลที่บกพร่องทางการได้ยินใช้ติดต่อสื่อสารกับบุคคลทั่วไปในสังคมชาวอเมริกันและชาวแคนาดาบางส่วน แตกต่างจากภาษาพูดที่นิยมใช้กันทั่วไปอย่างแพร่หลายเป็นภาษาสากล[7]

เรื่องการใช้คำ

ภาษาอังกฤษ - Time files

ภาษามือ - TIME ZOOM

เรื่องไวยากรณ์

ภาษาอังกฤษ : Subject + Verb + Object + Time

ภาษามือ : Time + Object +Subject + Verb

เช่น English : Jeff went to the gym yesterday.

ASL : YESTERDAY GYM JEFF GO

English : The dog chased the ball.

ASL : BALL DOG CHASE

นอกจากนี้ไวยากรณ์ของภาษามืออเมริกันยังมีความยืดหยุ่นสูง

เช่น English : I am a doctor.

ASL : I DOCTOR , DOCTOR I and I DOCTOR I

ใช้ได้ทั้ง 3 แบบ แต่แบบสุดท้ายเป็นการเน้น

ภาษามืออเมริกันไม่มีคำนำหน้านาม อย่าง a, an, the ไม่มี verb to be แล้วก็ไม่ต้องมีการเปลี่ยนแปลงรูปคำตามกาลของประโยค

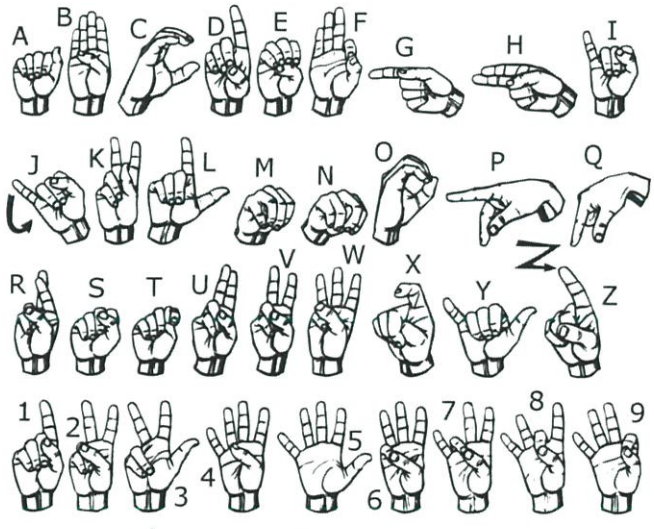
เช่น English – The cat ran up the tree yesterday.

ASL – YESTERDAY TREE CAT RUN-UP

นอกจากนี้ยังสามารถละประธานที่เป็น I ได้ เพราะว่า “ฉัน” เป็นคนพูด

English – I will cook eggs in the morning.

ASL – TOMORROW MORNING EGG COOK WILL

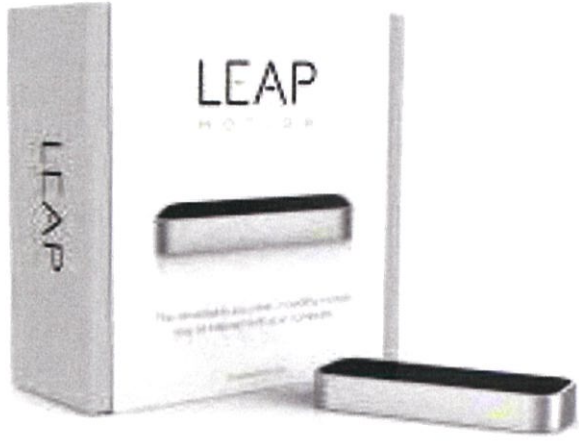


รูปที่ 2.2 ภาษามือแบบอเมริกัน (ASL)

ที่มา : http://coloringpagesjos.net/341983-sign-language-alphabet

2.4 Leap Motion 3D – Sensor

Leap Motion Controller เป็นอุปกรณ์เซนเซอร์สามมิติเชื่อมต่อด้วย USB ทำหน้าที่ตรวจจับการเคลื่อนไหวของมือหรือรูปทรงกระบอกต่างๆ มีความสามารถจับภาพ 300 เฟรมต่อวินาที มีความแม่นยำในระดับ 0.01 มิลลิเมตร



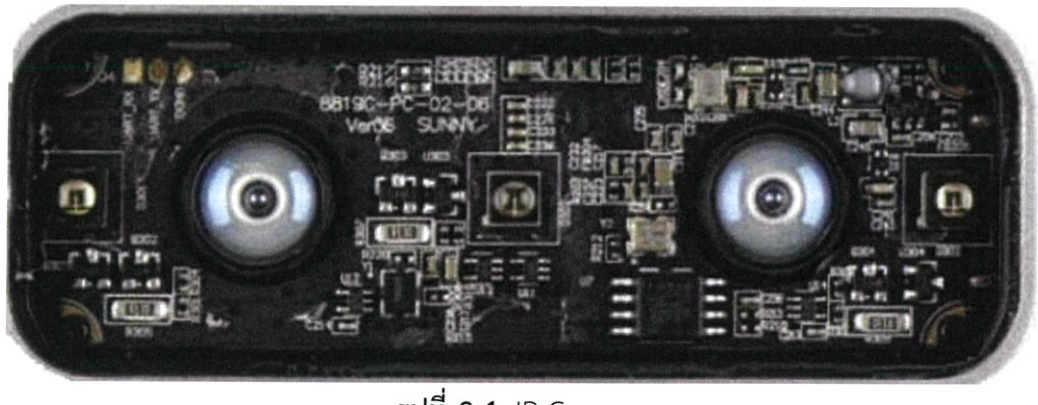
รูปที่ 2.3 อุปกรณ์ Leap Motion

ที่มา : http://www.amazon.co.uk/Leap-Motion-Controller-Interacts-Airspace/dp/B00C66Z9ZC

2.4.1 ส่วนประกอบภายในของ Leap Motion

2.4.1.1 กล้องอินฟราเรด

มีจำนวน 2 ตัว ใช้ในการรับภาพจากสิ่งของที่มาตกกระทบบนแสงอินฟราเรดที่ปล่อยออกมาและสะท้อนกลับมาที่ตัวอุปกรณ์ ทำให้เกิดภาพขึ้นมา

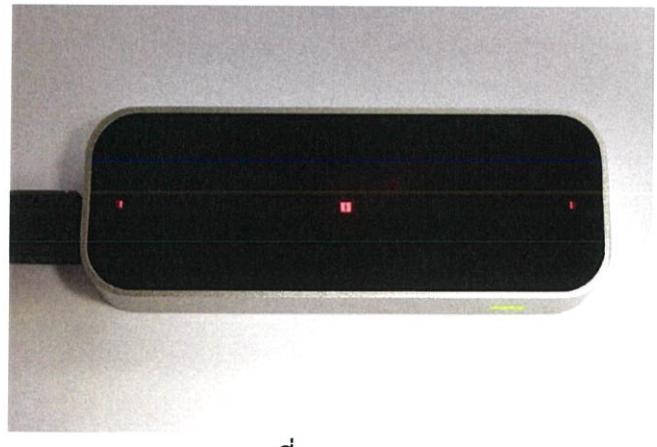


รูปที่ 2.4 IR Camera

ที่มา : http://ashlandtech.org/2014/04/02/product-comparison-kinect-and-leap-motion/

2.4.1.2 LED

มี LED จำนวน 3 ตัว บ่งบอกถึงการสามารถใช้งานได้ ณ ตอนนั้น



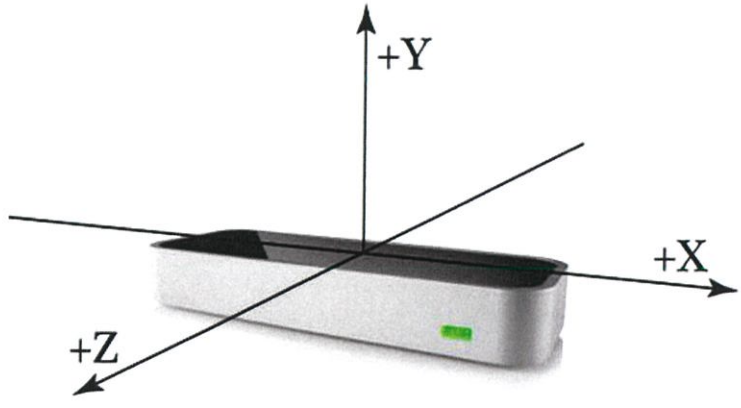
รูปที่ 2.5 LED

ที่มา : https://learn.sparkfun.com/tutorials/leap-motion-teardown

2.4.2 แกนอ้างอิงและระยะการตรวจจับ

2.4.2.1 แกนอ้างอิง

สามารถตรวจจับตำแหน่งของวัตถุต่างๆ เป็นสามมิติ อ้างอิงตามแกนแนวยาว (X) แกนความสูง (Y) และแกนแนวขวาง (Z) มีจุด origin อยู่ตรงกลางของตัวอุปกรณ์

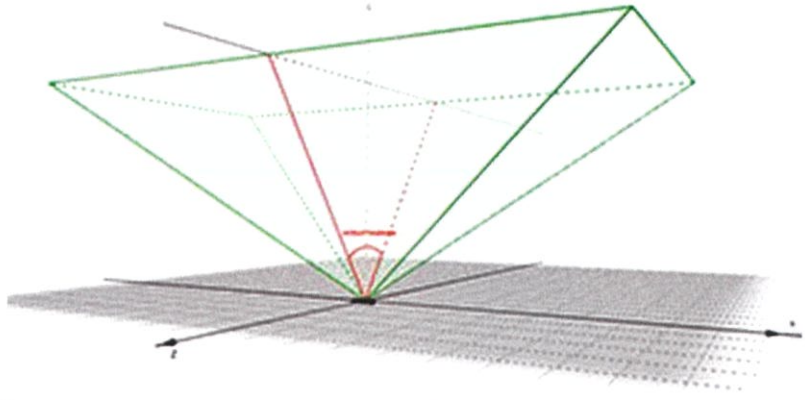


รูปที่ 2.6 แกนอ้างอิงของ Leap Motion

ที่มา : https://developer.leapmotion.com/getting-started/javascript/developer-guide

2.4.2.2 ระยะการตรวจจับ

มีระยะกว้าง 1 ฟุต รอบตัวอุปกรณ์แบบพีระมิด

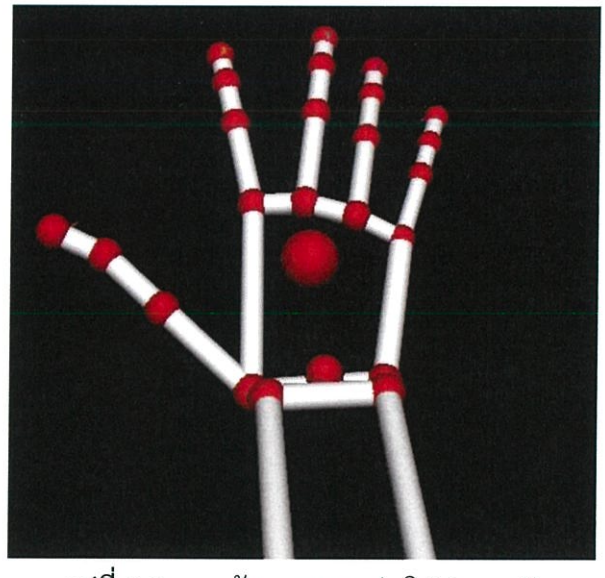


รูปที่ 2.7 ระยะการตรวจจับของ Sensor ในอุปกรณ์ Leap Motion
ที่มา : http://blog.peteshand.net/project/cba-product-demo/

2.4.3 การจับภาพ

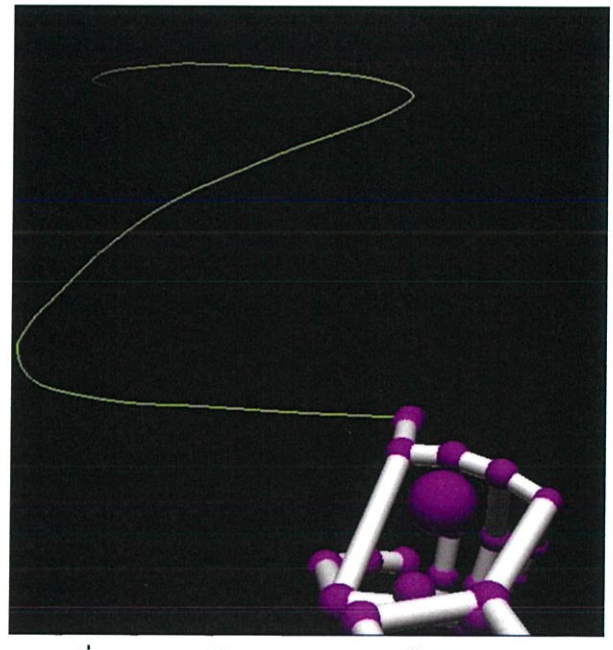
Leap motion มีการจับภาพ 3 แบบ คือ แบบปกติ แบบวาดหรือปัด และแบบวงกลม

2.4.3.1 แบบปกติ (Normal)



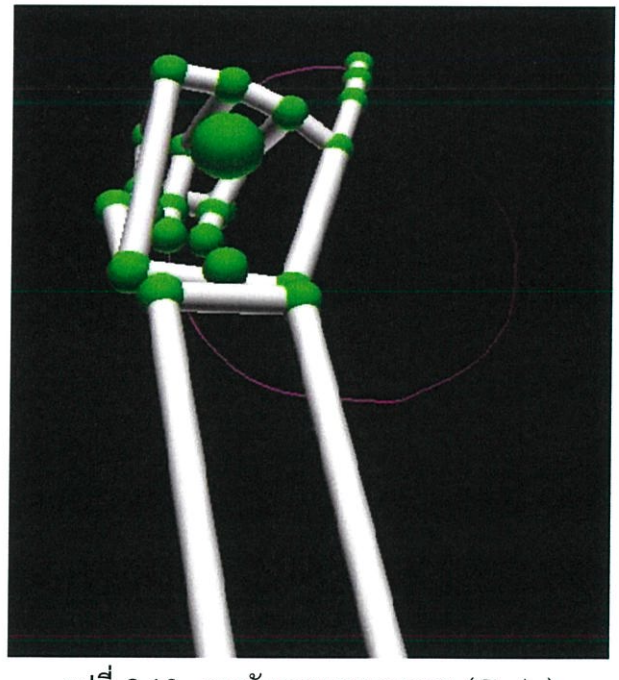
รูปที่ 2.8 การจับภาพแบบปกติ (Normal)

2.4.3.2 แบบวาดหรือปัด (Swipe)



รูปที่ 2.9 การจับภาพแบบวาดหรือปัด (Swipe)

2.4.3.3 แบบวงกลม (Circle)



รูปที่ 2.10 การจับภาพแบบวงกลม (Circle)

2.4.4 ขนาดอุปกรณ์และความต้องการขั้นต่ำ

2.4.4.1 ขนาดอุปกรณ์

สูง 0.5 นิ้ว กว้าง 1.2 นิ้ว ยาว 3 นิ้ว น้ำหนัก 0.1 ปอนด์ ภายในกล่องมีสาย USB ให้เลือกใช้ 2 ขนาดความยาว คือ 24 นิ้ว และ 60 นิ้ว Port ที่เชื่อมตัวฝั่ง Leap Motion เป็น Port USB แบบ microUSB 3.0

2.4.4.2 ความต้องการขั้นต่ำ

Window 7 หรือ 8 หรือ Mac Os X 10.7 Lion CPU AMD Phenom หรือ Intel core i3, i5, i7 Ram 2 GB USB 2.0 Port และ Internet

Height	**Width**	**Depth**	**Weight**
0.5 inches	1.2 inches	3 inches	0.1 pounds

Included Cables
24" and 60" USB 2.0 (microUSB 3.0 connectors)

Minimum System Requirements
Windows 7 or 8 or Mac OS X 10.7 Lion
AMD Phenom™ II or Intel® Core™ i3, i5, i7 processor
2 GB RAM
USB 2.0 port
Internet connection

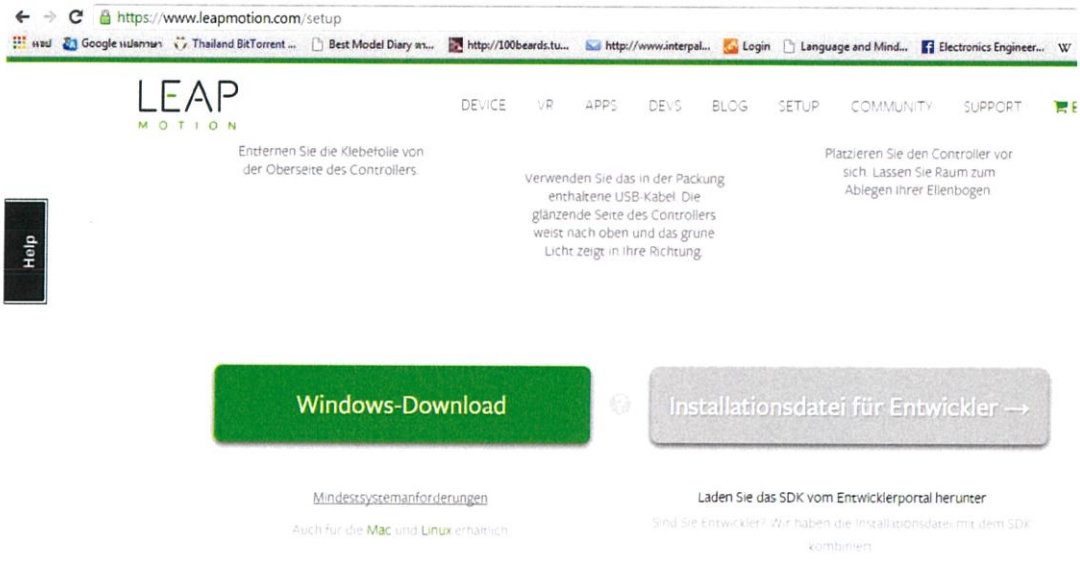
Warranty Terms
1 year limited

รูปที่ 2.11 ขนาดของอุปกรณ์ และความต้องการขั้นต่ำสำหรับการเชื่อมต่อ[6]
ที่มา : https://www.leapmotion.com/product

บทที่ 3
วิธีดำเนินการวิจัย

3.1 ติดตั้งโปรแกรมและระบบปฏิบัติการ

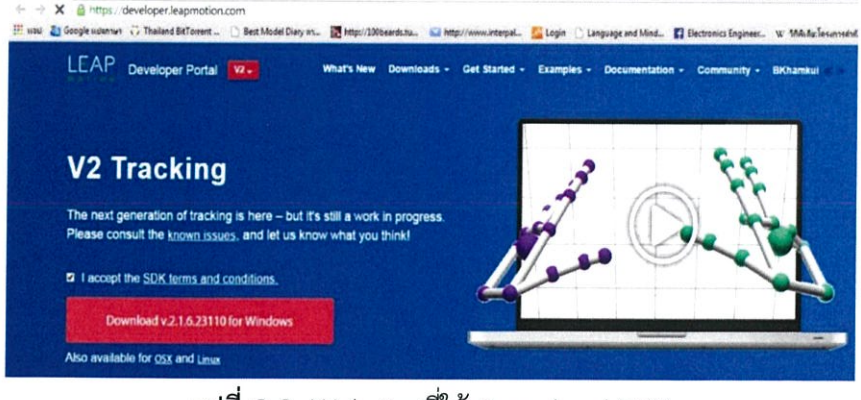
ติดตั้ง Leap Motion และ Software Development Kit (SDK) ของ Leap Motion ทำการ Download ตัว Set Up จาก https://www.leapmotion.com/setup



รูปที่ 3.1 Website ที่ใช้ Download Leap Motion software
ที่มา : https://www.leapmotion.com/setup

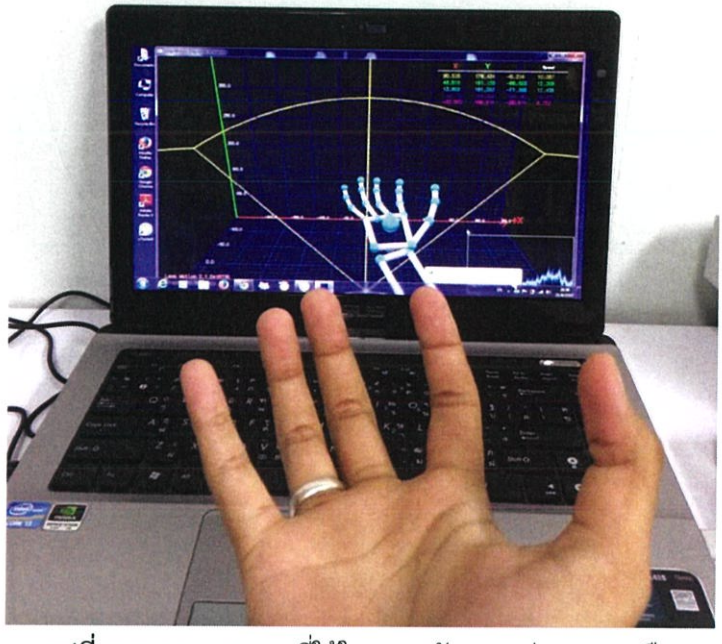
ขั้นตอนวิธีการติดตั้งโปรแกรม

1) ทำการ Download ตัว SDKจาก https://developer.leapmotion.com/ โดยต้องทำการสมัครสมาชิกก่อน



รูปที่ 3.2 Website ที่ใช้ Download SDK
ที่มา : https://developer.leapmotion.com/

2) ทำการติดตั้ง SDK และเชื่อมตัว Leap Motion เข้ากับ Laptopพร้อมทั้งเปิด Visualizer ใน SDK

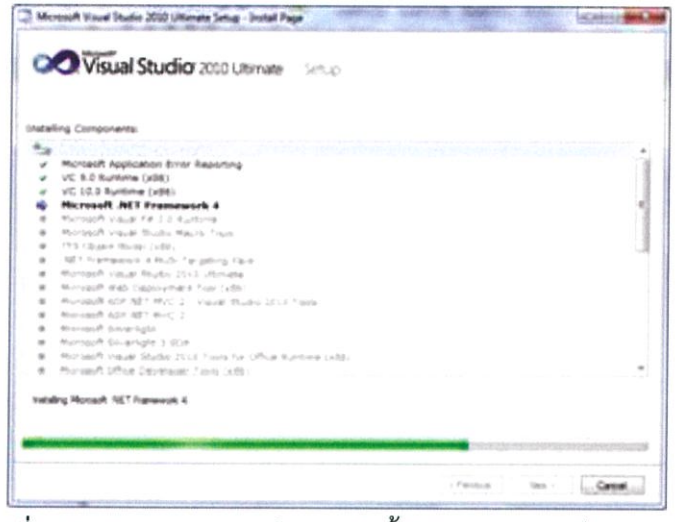


รูปที่ 3.3 Visualizer ที่ใช้ในการดูลักษณะท่าทางของมือ

3) ติดตั้งโปรแกรม Microsoft Visual Studio 2010

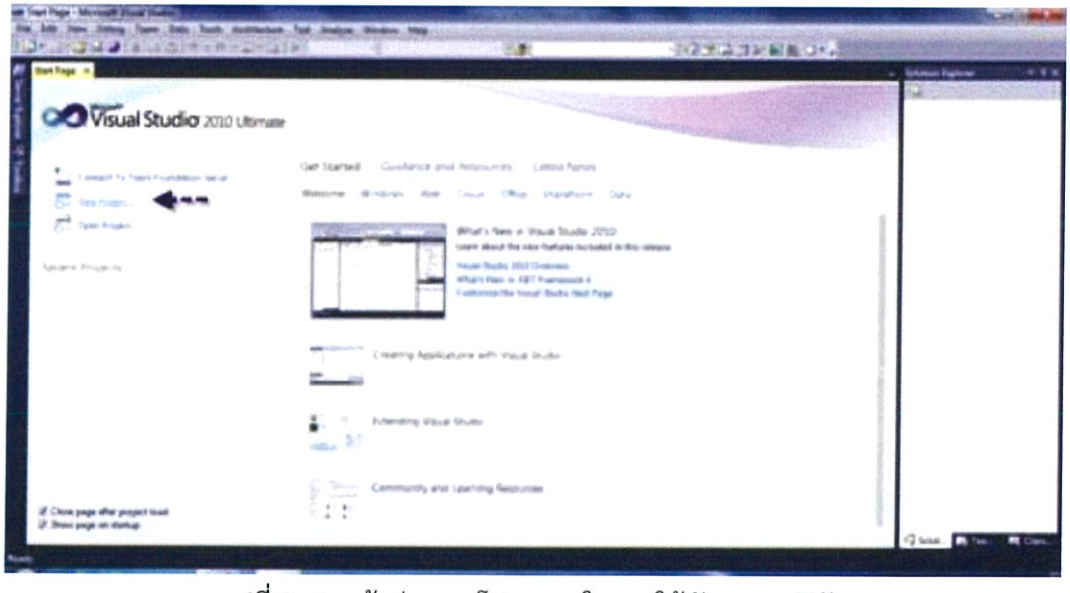
รูปที่ 3.4 การติดตั้งโปรแกรม Microsoft Visual Studio 2010

4) โปรแกรมจะทำการติดตั้งจนเสร็จสมบูรณ์



รูปที่ 3.5 Components ในการติดตั้งโปรแกรมจนเสร็จสมบูรณ์

5) เมื่อติดตั้งเสร็จจะได้รูปร่างหน้าตาของโปรแกรมที่ใช้พัฒนาระบบดังต่อไปนี้

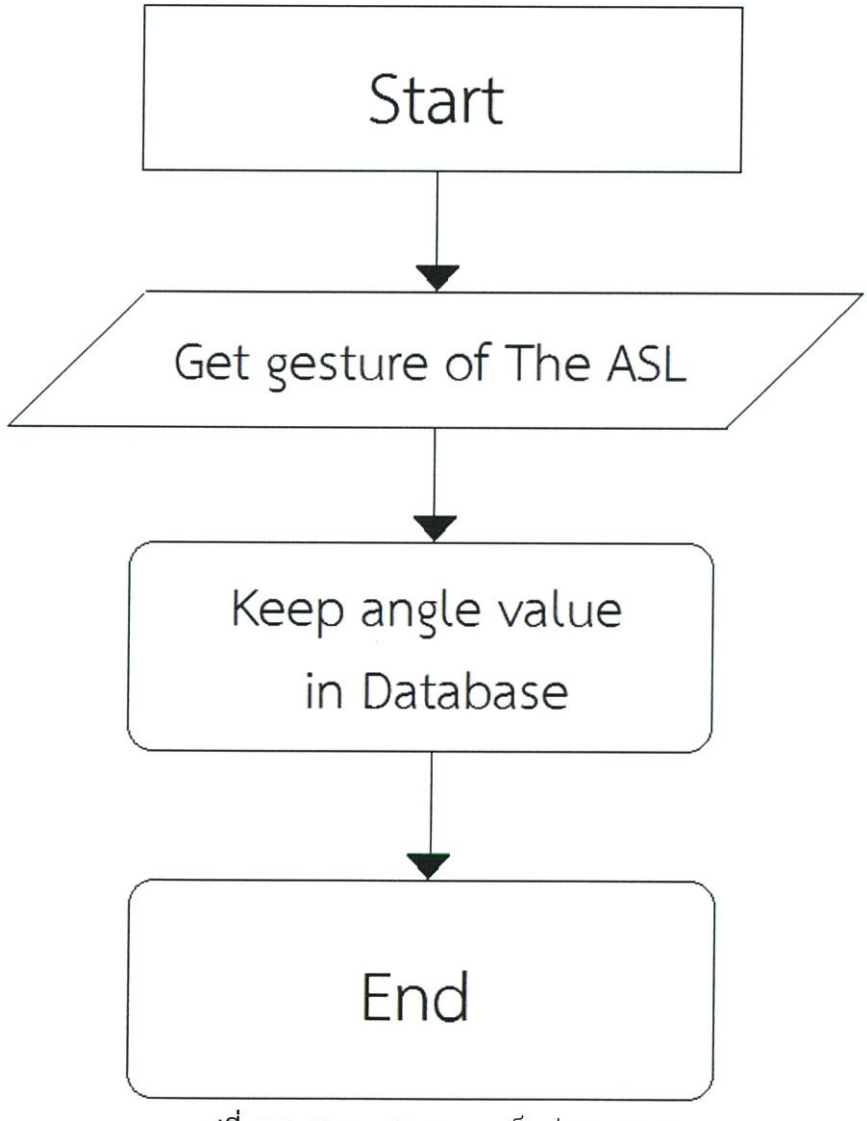


รูปที่ 3.6 หน้าต่างของโปรแกรมในการใช้พัฒนาการวิจัย

3.2 กระบวนการทำงาน

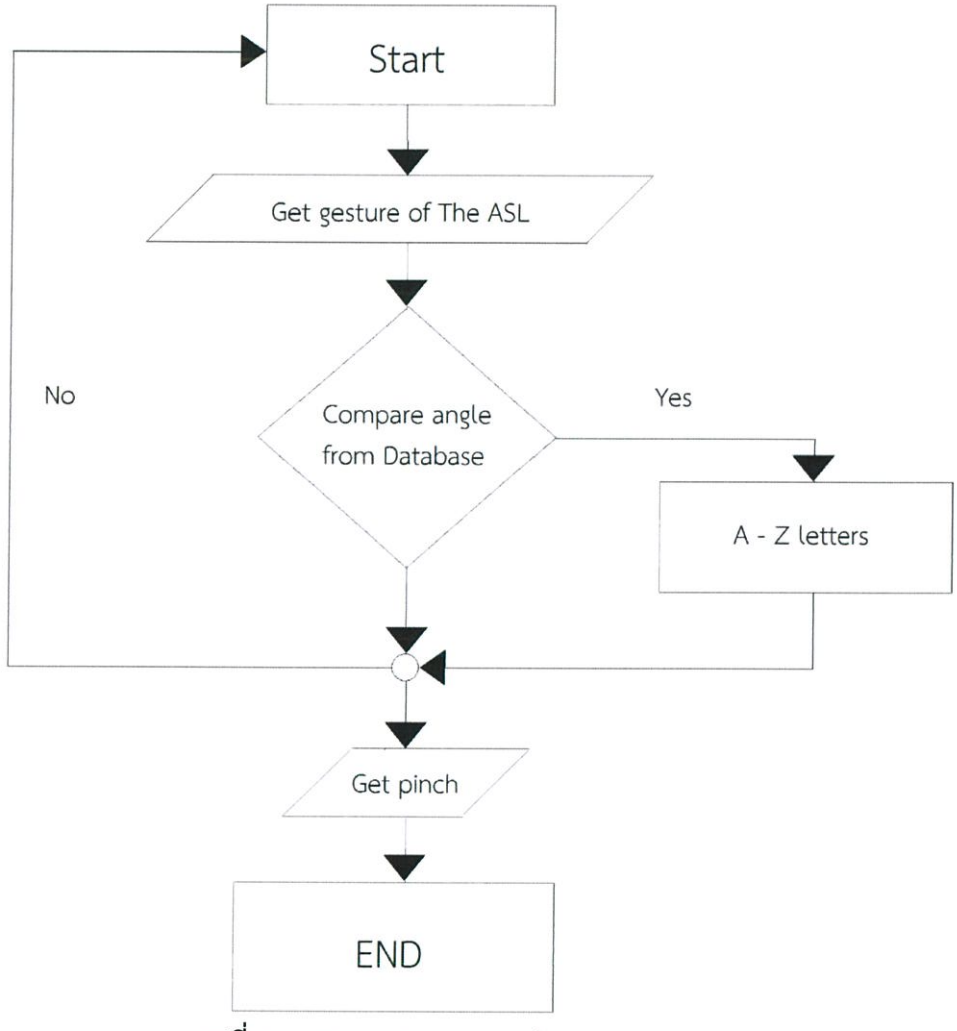
ลักษณะการดำเนินงานมี 2 ส่วน เก็บข้อมูล และ แปลภาษา

1) เริ่มต้นจากผู้ใช้งานแสดงท่าทางเป็นตัวอักษรภาษาอังกฤษของภาษามือแบบอเมริกันเหนือตัวอุปกรณ์ Leap Motion เก็บค่ามุมของข้อนิ้วของแต่ละตัวอักษรลงใน Database ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 Flow chart การเก็บค่าของระบบ

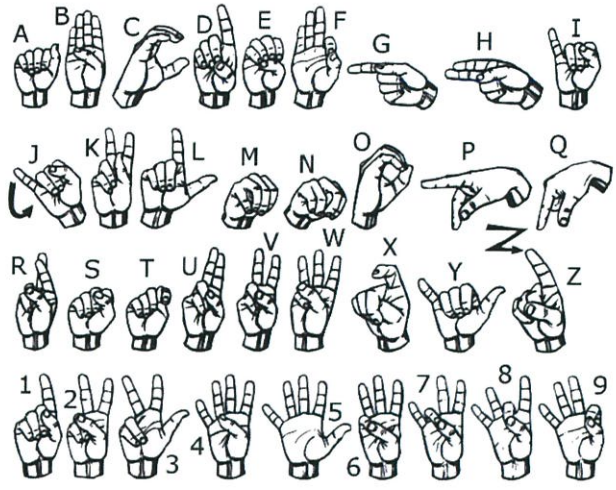
2) เริ่มต้นจากผู้ใช่งานแสดงท่าทางเป็นตักษรภาษาอังกฤษของภาษามือแบบอเมริกันเหนือตัวอุปกรณ์ Leap Motion เมื่อเปรียบเทียบกับข้อมูลที่เก็บจากส่วนที่ 1 ถ้าตรงตามเงื่อนไขจะแสดงตัวอักษร ถ้าไม่ระบบจะกลับไปให้แสดงท่าทางใหม่ หากต้องการหยุดระบบ ให้บีบนิ้วหัวชี้เข้าหาหัวแม่มือ



รูปที่ 3.8 Flow chart การแปลภาษาของระบบ

3.2.1 การแสดงท่าทางของตัวอักษรภาษาอังกฤษ

จะเป็นการใช้ภาษามือแสดงท่าทางแทนตัวอักษรภาษาอังกฤษ ทั้งหมด 26 ตัวอักษร โดยแต่ละอักษรจะมีท่าทางที่แตกต่างกัน ดังนี้

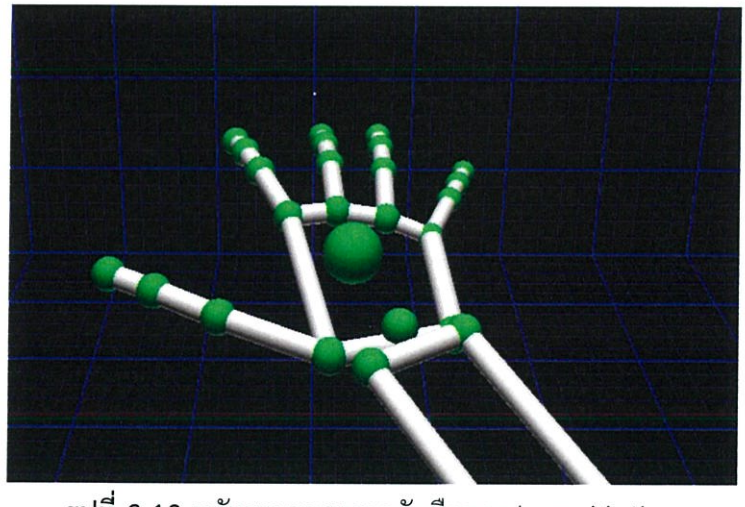


รูปที่ 3.9 ภาษามือแบบอเมริกัน (ASL)

ที่มา : http://coloringpagesjos.net/341983-sign-language-alphabet

3.2.2 กระบวนการตรวจจับของ Leap Motion

หลังจาก ทำการแสดงท่าทางของตัวอักษรบนอุปกรณ์ Leap Motion อุปกรณ์จะทำการตรวจจับข้อต่อของกระดูกของมือซึ่งประกอบด้วย กระดูกนิ้วมือ กระดูกฝ่ามือ และกระดูกแขนท่อนปลาย



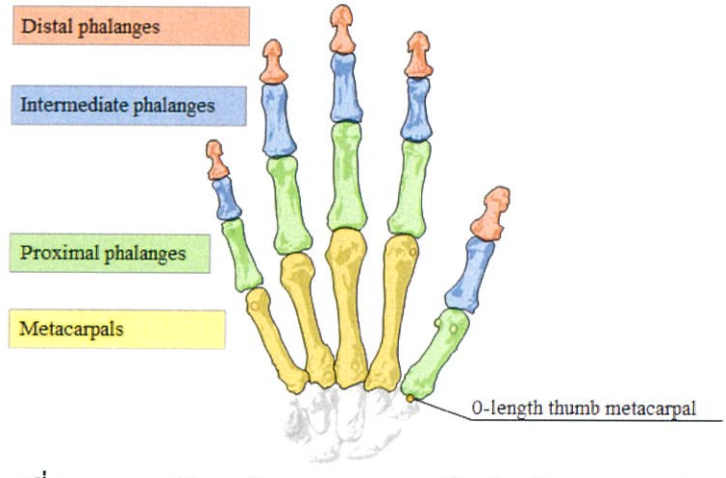
รูปที่ 3.10 หลักฐานการตรวจจับมือของ Leap Motion

3.2.3 คอมพิวเตอร์ทำการประมวลผล

ตารางที่ 3.1 คุณลักษณะของคอมพิวเตอร์ที่ใช้ในการทดลอง

รายการ	คุณลักษณะ
CPU	Core i3 – 2310M 2.1 GHz
Memory	4.00 GB RAM
Operation System	Window 7 Ultimate รุ่น 64 - Bit

3.2.4 กระบวนการทางคณิตศาสตร์เพื่อหาค่ามุม



รูปที่ 3.11 กายวิภาคจำลองของกระดูกมือ สำหรับ Leap Motion
ที่มา : http://blog.leapmotion.com/skeletal-tracking-101-getting-started-with-the-bone-api-and-rigged-hands/

จากรูปที่ 3.11 มือประกอบด้วยนิ้วจำนวน 5 นิ้ว แต่ละนิ้ว มีกระดูกนิ้วจำนวน 4 ชิ้น ยกเว้นนิ้วหัวแม่มือที่มีแค่ 3 นิ้วโดยกระดูกนิ้วมือแต่ละส่วนมีชื่อว่า

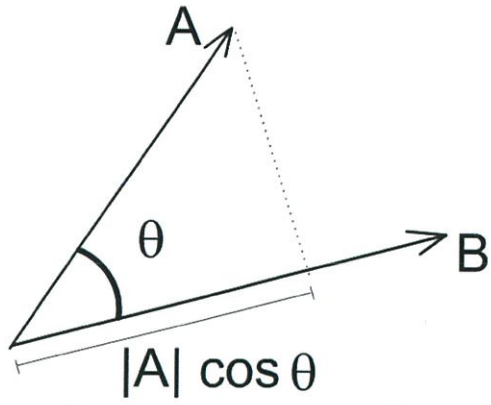
- Distal phalanges
- Intermediate phalanges
- Proximal phalanges
- Metacarpals

3.2.4.1 Dot product

การหามุมสามารถทำได้โดยวิธี Dot product นิยามของ Dot product[2]

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \qquad (3.1)$$

แต่ในทางเรขาคณิต



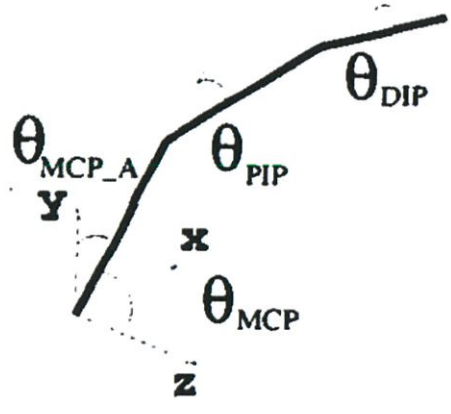
รูปที่ 3.12 การ Dot product เพื่อหามุม
ที่มา : http://en.wikipedia.org/wiki/Dot_product

ในปริภูมิแบบยุคลิด ผลคูณไขว้มีความสัมพันธ์กับความยาวและมุม สำหรับเวกเตอร์ a ผลลัพธ์ของ

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}|\,|\mathbf{b}| \cos\theta \qquad (3.2)$$

โดยที่ a · a คือกำลังสองของความยาวของ a

ส่วนในกรณีทั่วไปเมื่อนำ b มาใช้กับมุมคือข้อนิ้วของแต่ละนิ้ว สามารถแสดงได้ดังรูปที่ 3.13



รูปที่ 3.13 มุมของข้อนิ้ว

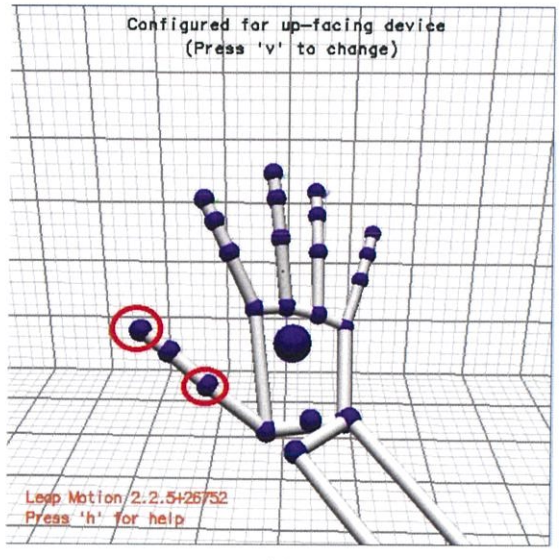
3.2.4.2 ตัวแปรการใช้งาน

โดยทุกนิ้วจะหามุมได้โดยการ

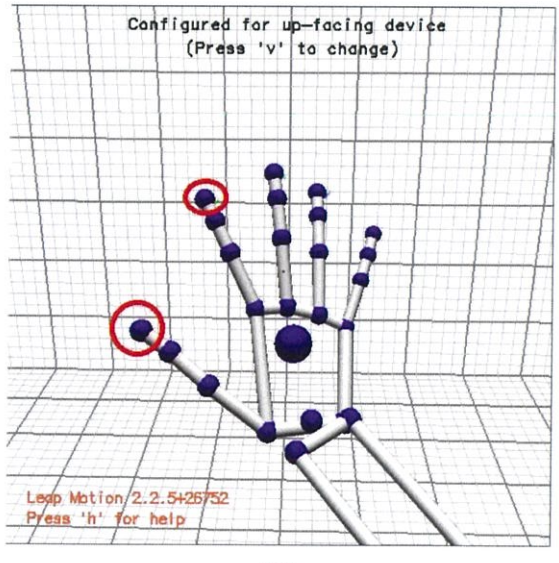
- Dot กันระหว่าง Distal กับ Intermediate
- Dot กันระหว่าง Intermediate กับ Proximal
- Dot กันระหว่าง Proximal กับ Metacarpals

หามุมระหว่างข้อนิ้วระหว่างนิ้ว

- Dot กันระหว่าง Proximal
- Dot กันระหว่าง Distal



(ก)



(ข)

รูปที่ 3.14 (ก) Dot กันระหว่างข้อนิ้วภายในนิ้วเดียวกัน
(ข) Dot กันระหว่างข้อนิ้วของนิ้วอื่น

3.2.5 Source code และการแสดงผลด้วยโปรแกรม Visual Studio

3.2.5.1 การกำหนดตัวแปรและตัวอย่าง code

ด้วยการใช้ภาษา C++ ในการเขียน Code เขียน Code ของ Leap Motion สามารถดูได้จาก API ใน Folder ของ SDK ที่ Download มาจาก Website ของ Leap Motion โดยทำการ Set parameter เพื่อกำหนดค่ามุมของแต่ละนิ้ว ดังนี้

A0 = ไม่มีค่าเนื่องจาก Metacarpals ของ นิ้วหัวแม่มือมีค่าเป็น 0

A1 = Dot product ระหว่าง Proximal กับ Metacarpals ของ นิ้วชี้

A2 = Dot product ระหว่าง Proximal กับ Metacarpals ของ นิ้วกลาง

A3 = Dot product ระหว่าง Proximal กับ Metacarpals ของ นิ้วนาง

A4 = Dot product ระหว่าง Proximal กับ Metacarpals ของ นิ้วก้อย

B0 = Dot product ระหว่าง Intermediate กับ Proximal ของ นิ้วหัวแม่มือ

B1 = Dot product ระหว่าง Intermediate กับ Proximal ของ นิ้วชี้

B2 = Dot product ระหว่าง Intermediate กับ Proximal ของ นิ้วกลาง

B3 = Dot product ระหว่าง Intermediate กับ Proximal ของ นิ้วนาง

B4 = Dot product ระหว่าง Intermediate กับ Proximal ของ นิ้วก้อย

C0 = Dot product ระหว่าง Distal กับ Intermediate ของ นิ้วหัวแม่มือ

C1 = Dot product ระหว่าง Distal กับ Intermediate ของ นิ้วชี้

C2 = Dot product ระหว่าง Distal กับ Intermediate ของ นิ้วกลาง

C3 = Dot product ระหว่าง Distal กับ Intermediate ของ นิ้วนาง

C4 = Dot product ระหว่าง Distal กับ Intermediate ของ นิ้วก้อย

AA1 = Dot product ระหว่าง Proximal ของนิ้วชี้และนิ้วกลาง

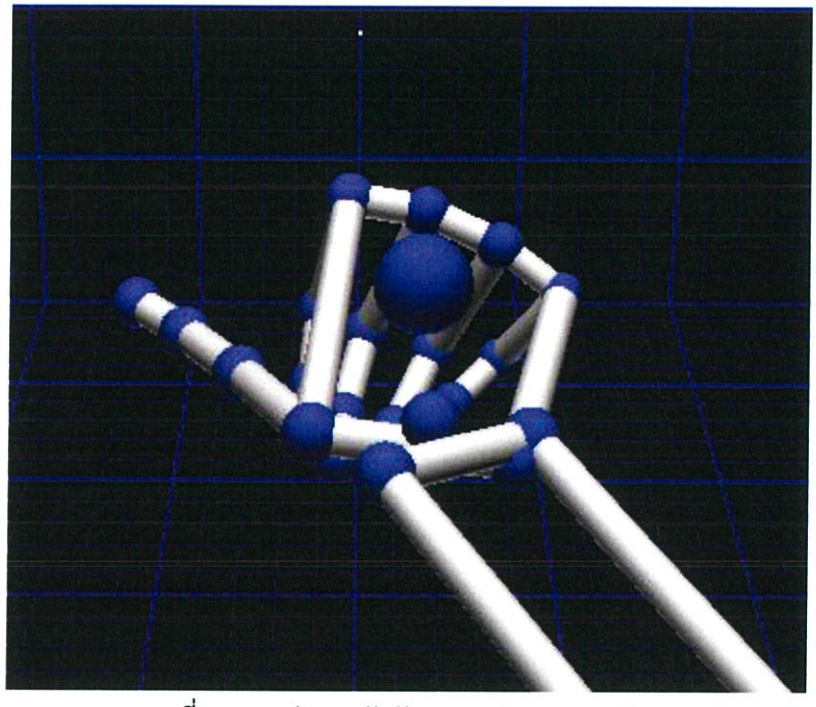
BB1 = Dot product ระหว่าง Proximal ของนิ้วกลางและนิ้วนาง

CC1 = Dot product ระหว่าง Proximal ของนิ้วนางและนิ้วก้อย

DD1 = Dot product ระหว่าง Distal ของนิ้วหัวแม่มือและนิ้วชี้

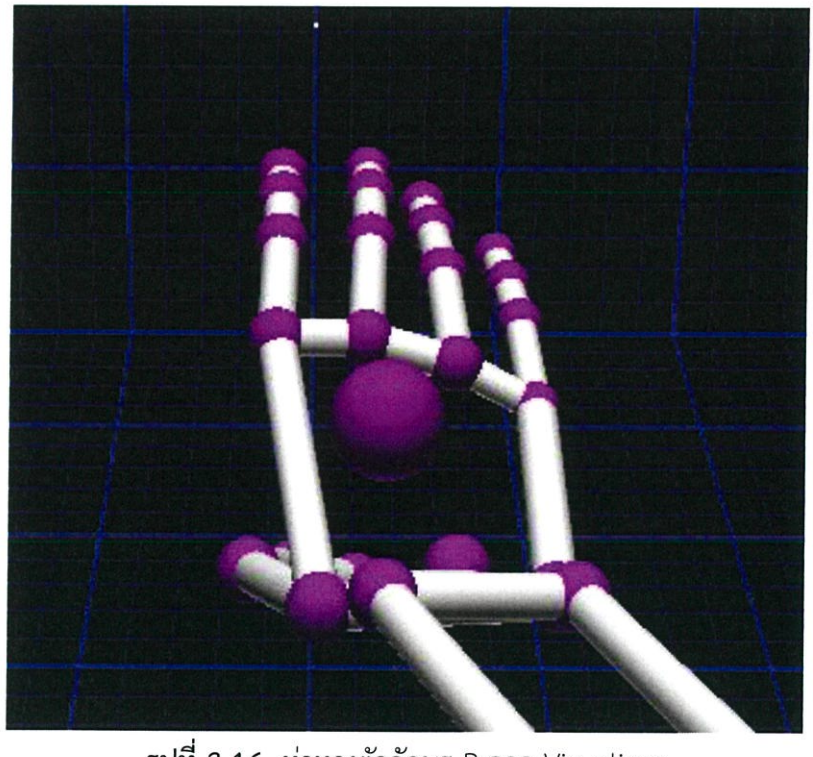
3.2.5.2 การทำท่าทางตัวอักษรต่างๆ

ตัวอักษร A กำมือและนำนิ้วหัวแม่มือเปิดออกลักษณะคล้ายการโบกรถ



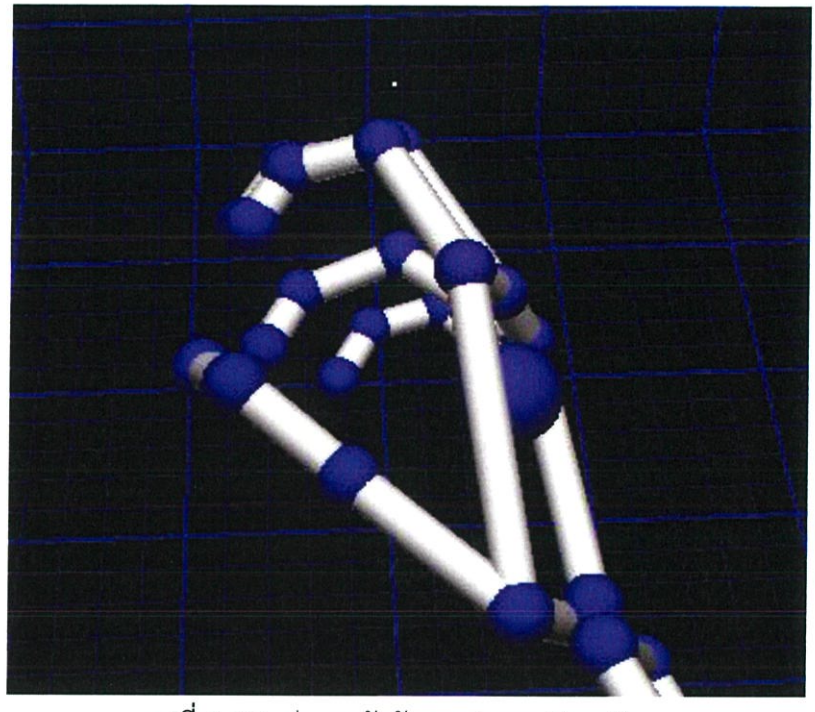
รูปที่ 3.15 ท่าทางตัวอักษร A จาก Visualizer

ตัวอักษร B คว่ำมือและนำนิ้วหัวแม่มือหุบเข้าไปคล้ายการแสดงสัญลักษณ์เลข 4



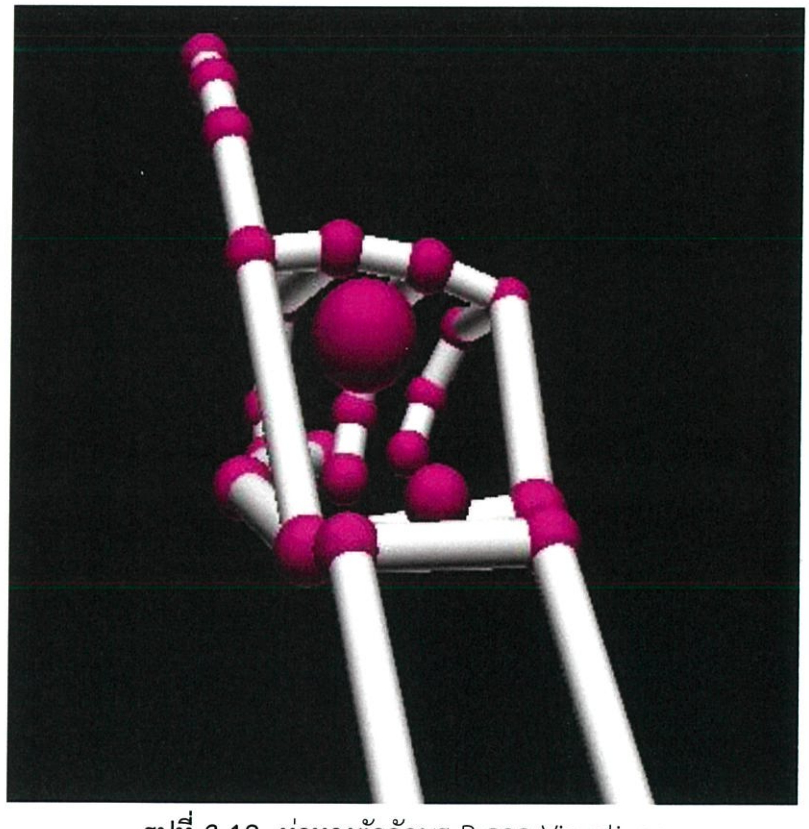
รูปที่ 3.16 ท่าทางตัวอักษร B จาก Visualizer

ตัวอักษร C งอนิ้วมือทั้ง 5 ทำเป็นรูปตัว C



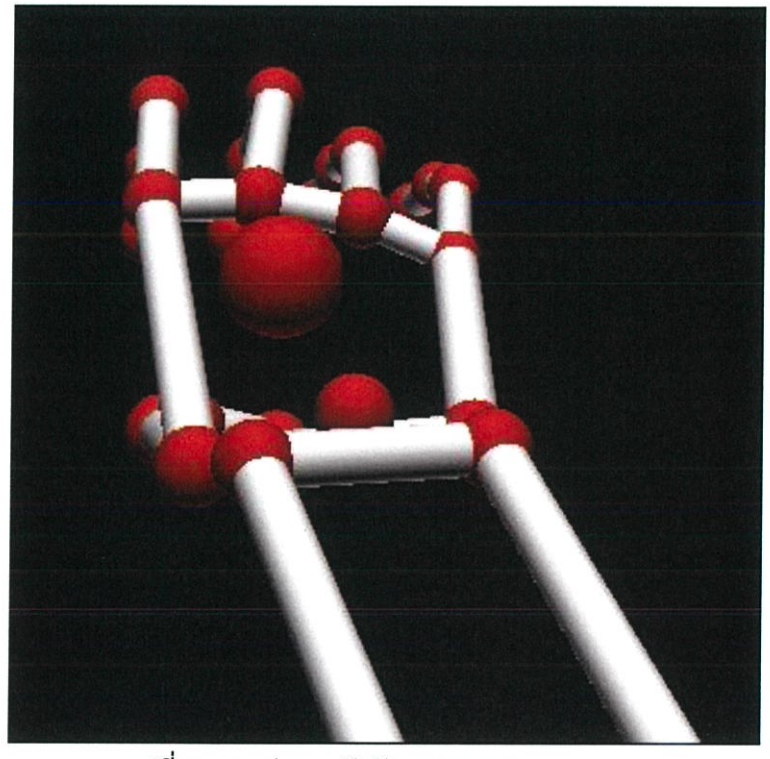
รูปที่ 3.17 ท่าทางตัวอักษร C จาก Visualizer

ตัวอักษร D นิ้วหัวแม่มือประกบระหว่างนิ้วกลางและนิ้วนาง นิ้วชี้ เหยียดออก



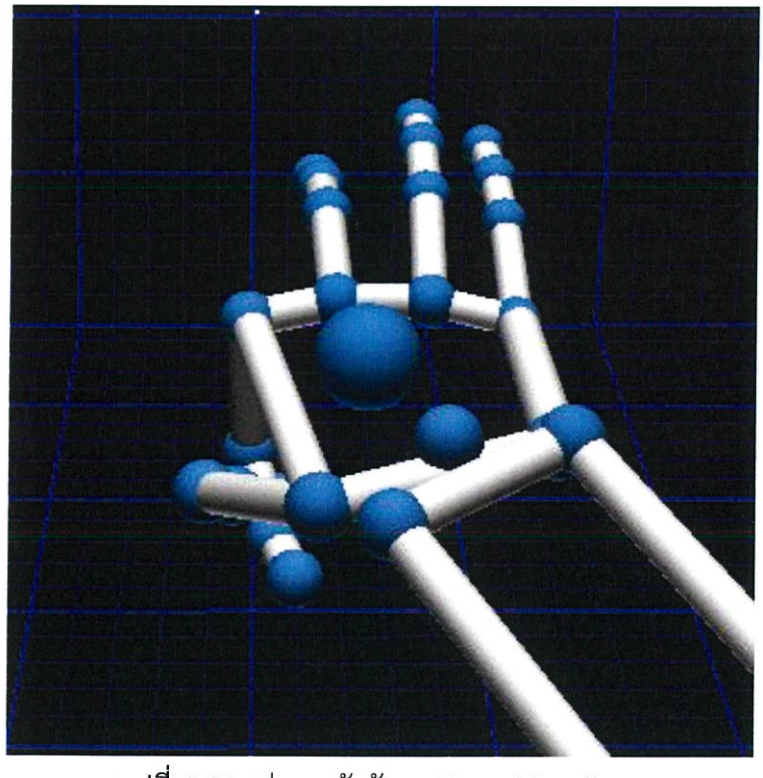
รูปที่ 3.18 ท่าทางตัวอักษร B จาก Visualizer

ตัวอักษร E งอนิ้วทั้ง 5 นิ้วเข้าหากัน



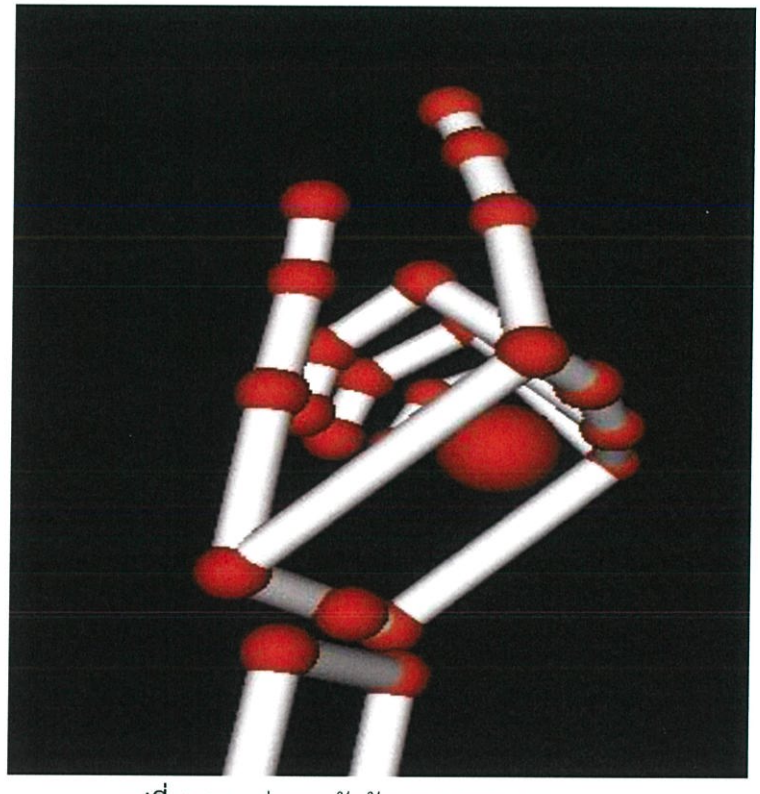
รูปที่ 3.19 ท่าทางตัวอักษร E จาก Visualizer

ตัวอักษร F นิ้วหัวแม่มือกับนิ้วชี้ผสานกันเป็นวงกลม 3 นิ้วที่เหลือ เหยียดออก



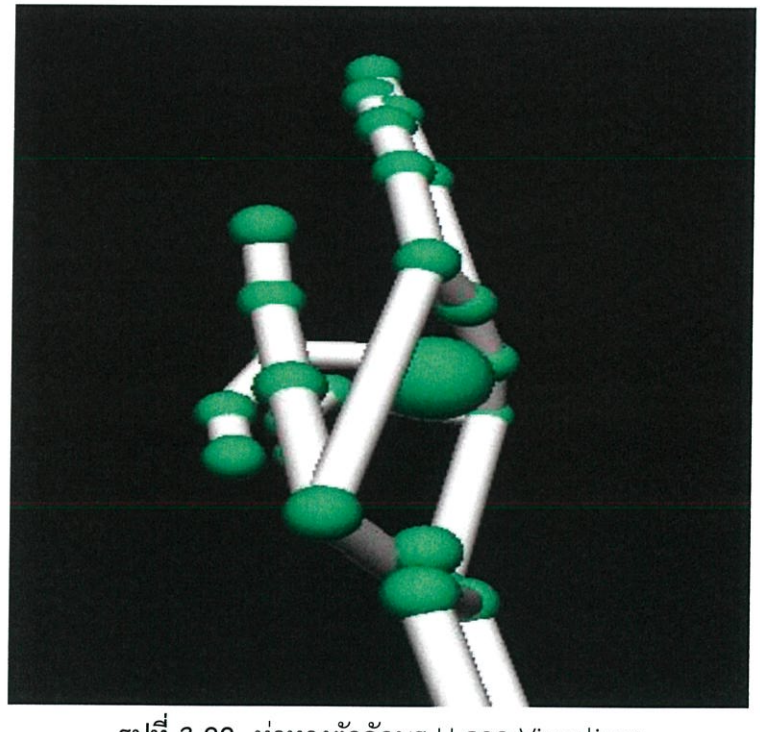
รูปที่ 3.20 ท่าทางตัวอักษร F จาก Visualizer

ตัวอักษร G นิ้วหัวแม่มือกับนิ้วชี้ ทำท่าคีบแคบๆ นิ้วอื่นๆ กำแน่น



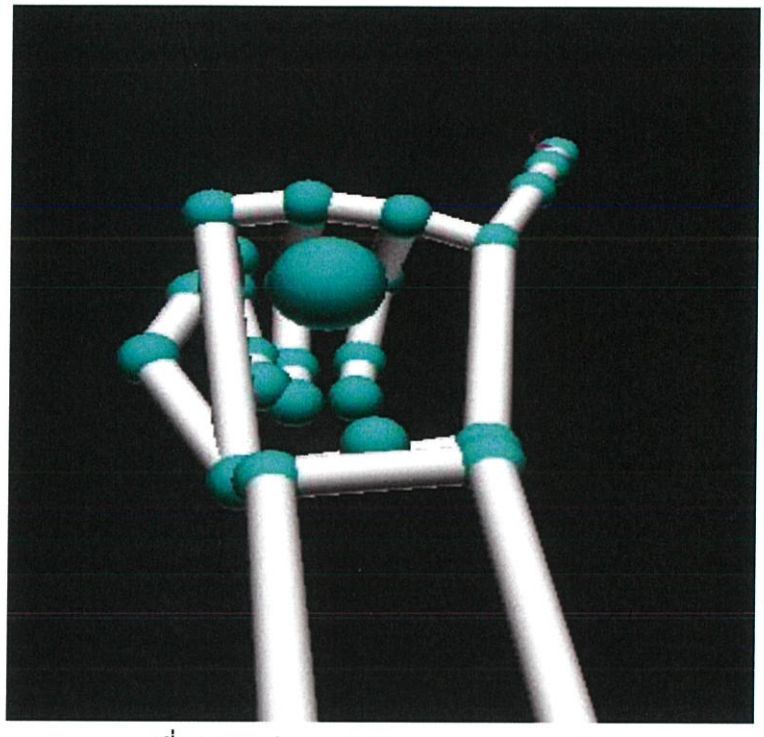
รูปที่ 3.21 ท่าทางตัวอักษร G จาก Visualizer

ตัวอักษร H ชู 2 นิ้วเอียงไปด้านข้าง นิ้วหัวแม่มือวางบนนิ้วนาง



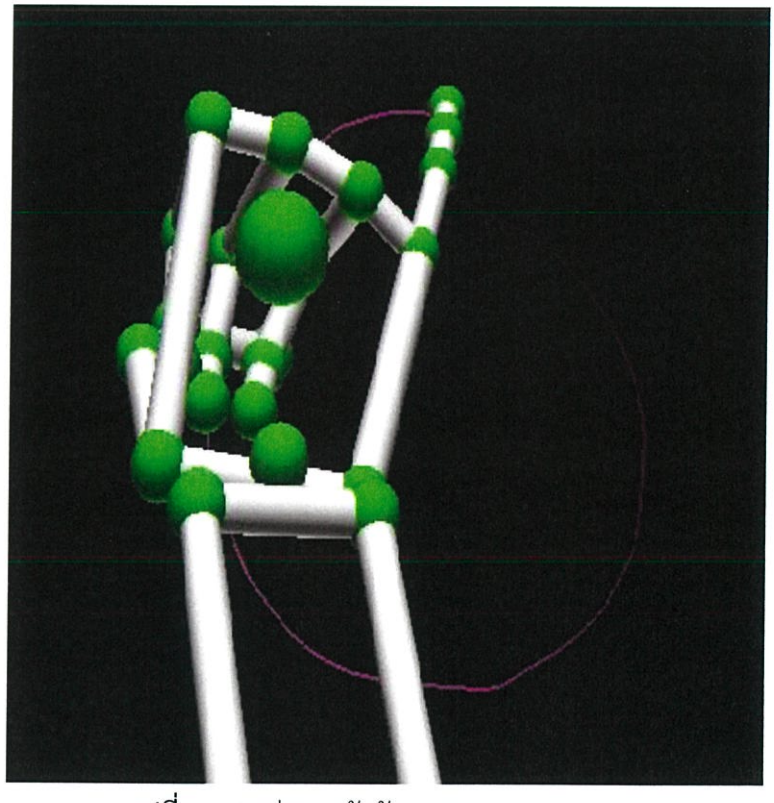
รูปที่ 3.22 ท่าทางตัวอักษร H จาก Visualizer

ตัวอักษร I กำมือ เหยียดนิ้วก้อย



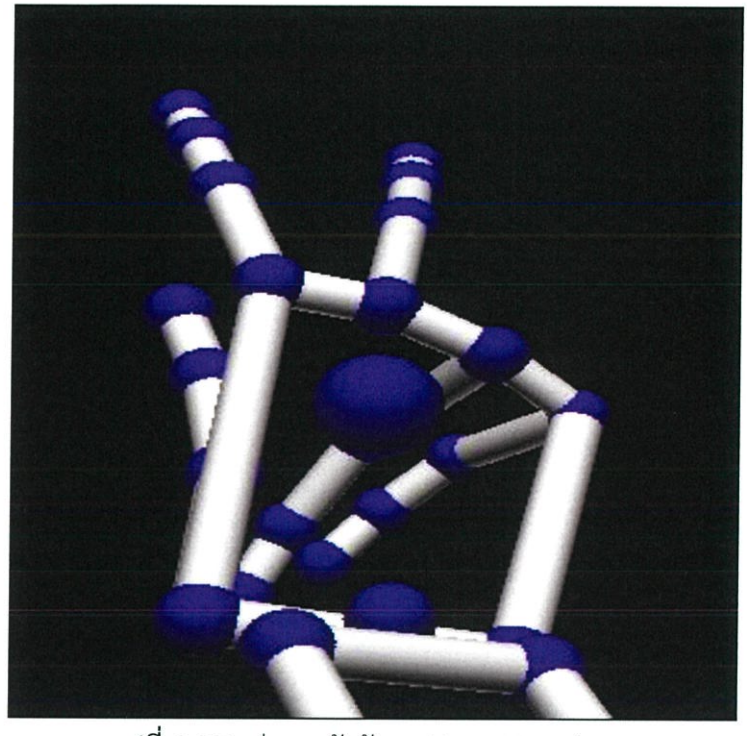
รูปที่ 3.23 ท่าทางตัวอักษร I จาก Visualizer

ตัวอักษร J ท่าทางเหมือนตัวอักษร I แต่นำนิ้วก้อยวนเป็นรูปวงกลม



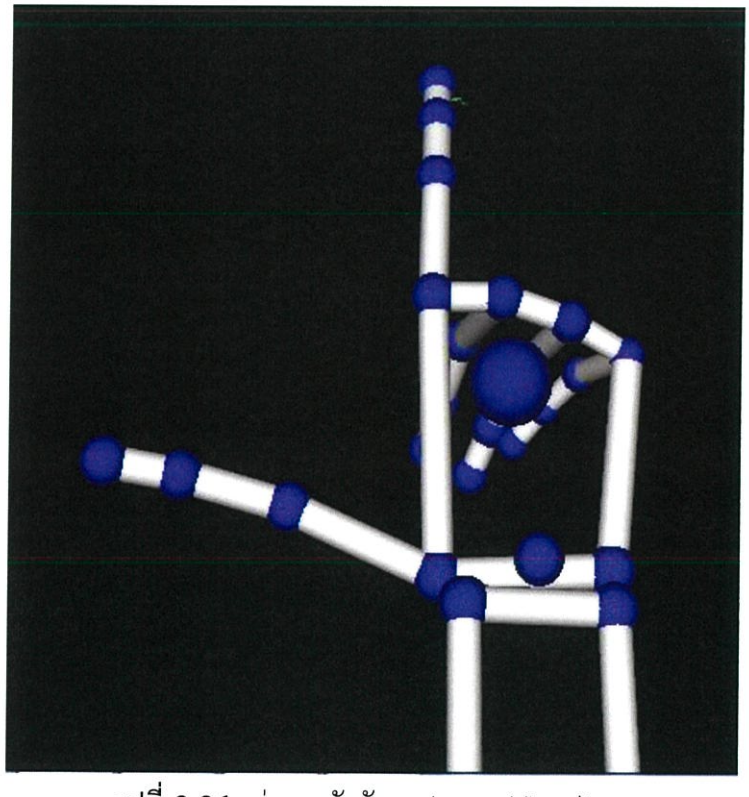
รูปที่ 3.24 ท่าทางตัวอักษร J จาก Visualizer

ตัวอักษร K ชูสองนิ้ว นิ้วหัวแม่มือแนบติดระหว่างนิ้วชี้และนิ้วกลาง



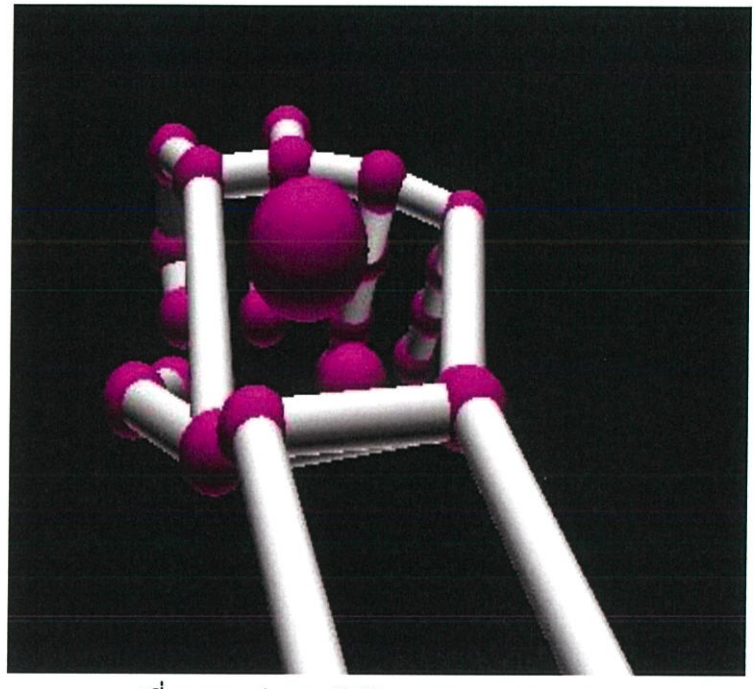
รูปที่ 3.25 ท่าทางตัวอักษร K จาก Visualizer

ตัวอักษร L กำมือ กางนิ้วหัวแม่มือและนิ้วชี้ เป็นรูปร่างคล้ายตัว L



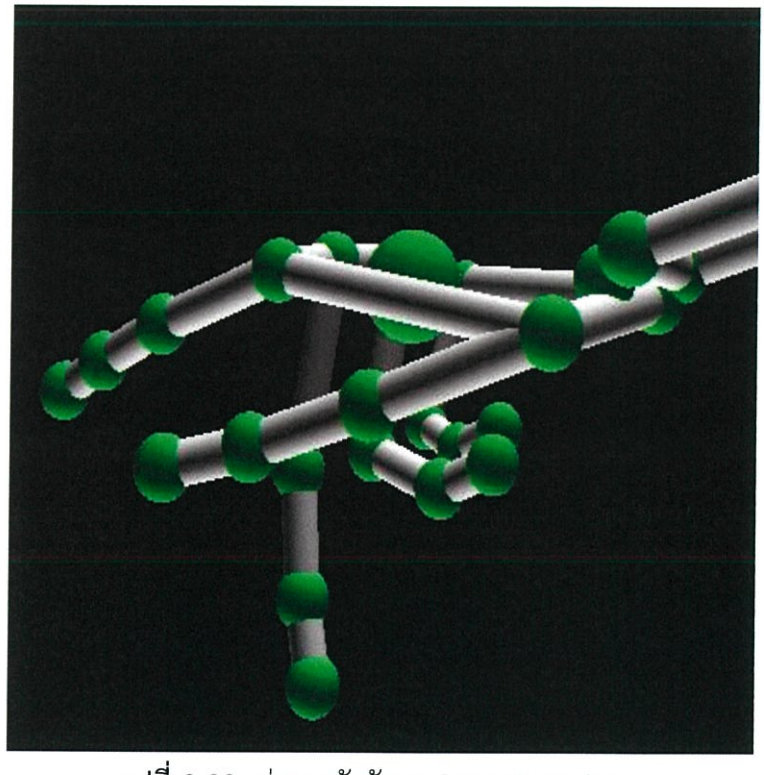
รูปที่ 3.26 ท่าทางตัวอักษร L จาก Visualizer

ตัวอักษร O กำมือแน่น



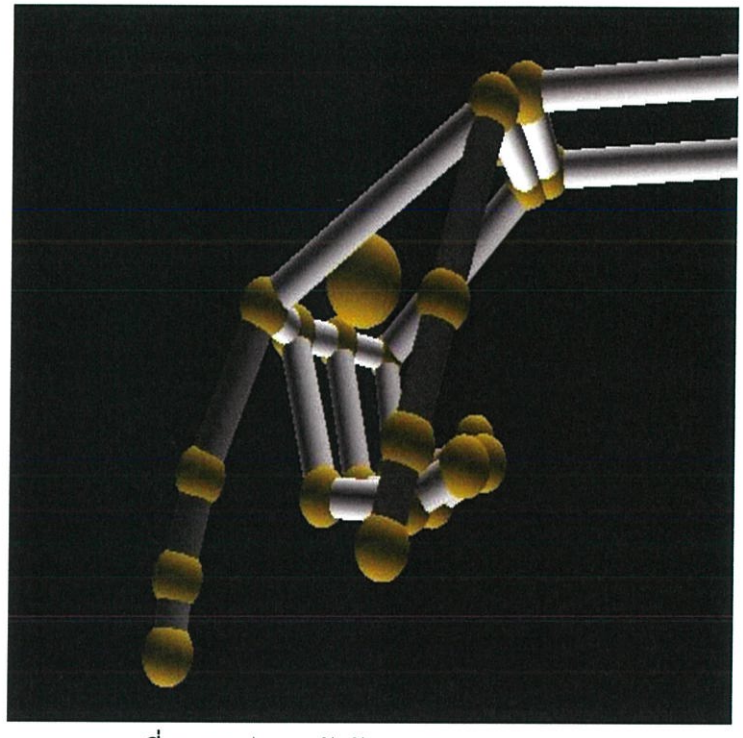
รูปที่ 3.27 ท่าทางตัวอักษร O จาก Visualizer

ตัวอักษร P กำมือ เหยียดนิ้วชี้ ทิ้งนิ้วกลางลง นิ้วหัวแม่มือพุ่งไปข้างหน้าแนบกับนิ้วกลาง



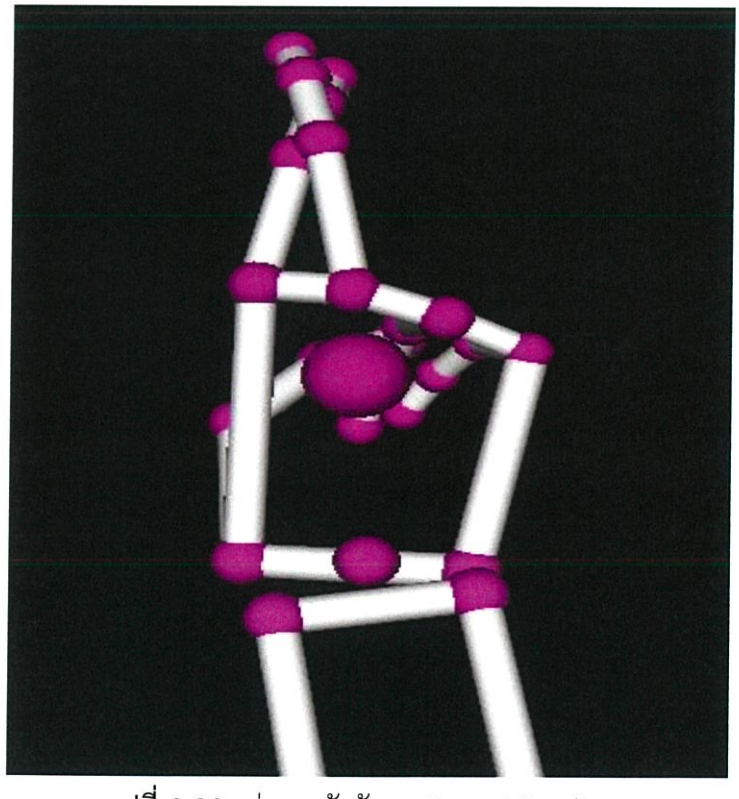
รูปที่ 3.28 ท่าทางตัวอักษร P จาก Visualizer

ตัวอักษร Q กำมือ นิ้วหัวแม่มือและนิ้วชี้ทำเหมือนคีบแต่กว้างกว่า ตัว G



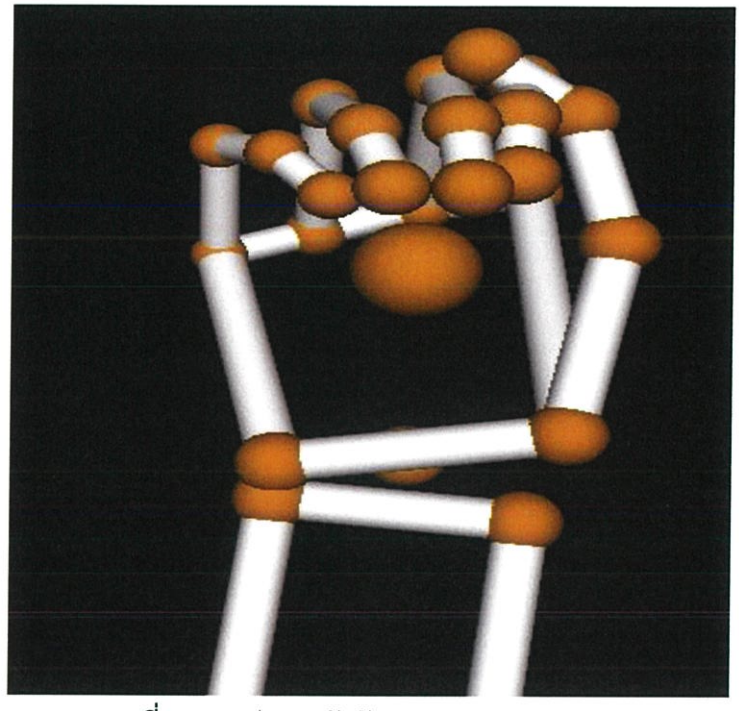
รูปที่ 3.29 ท่าทางตัวอักษร Q จาก Visualizer

ตัวอักษร R ชูสองนิ้ว ไขว้นิ้วกัน นิ้วกลางอยู่ด้านบน



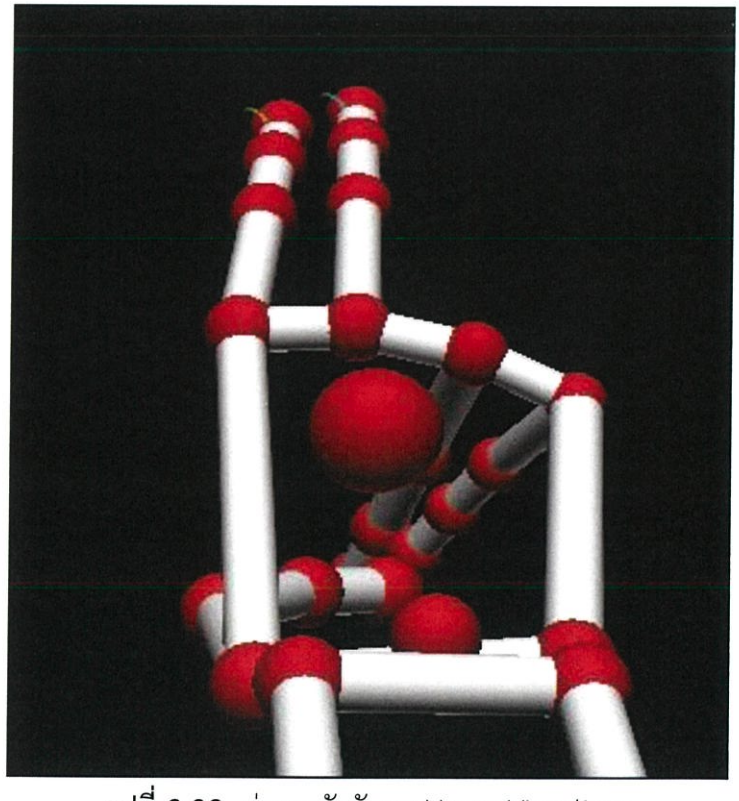
รูปที่ 3.30 ท่าทางตัวอักษร R จาก Visualizer

ตัวอักษร S กำมือนิ้วหัวแม่มือแนบนิ้วชี้แต่ไม่เกินออกมา



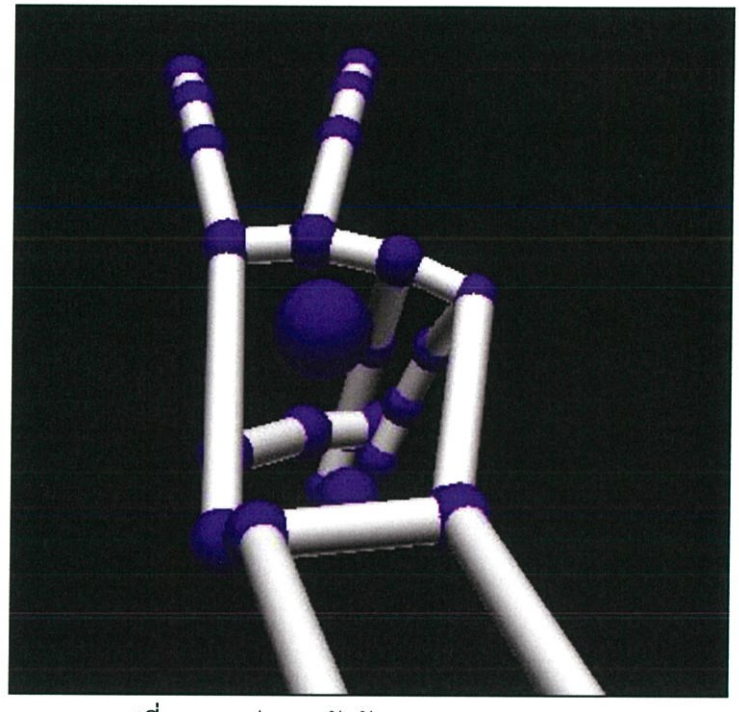
รูปที่ 3.31 ท่าทางตัวอักษร S จาก Visualizer

ตัวอักษร U ชูสองนิ้ว นิ้วชี้กับนิ้วกลางติดกัน



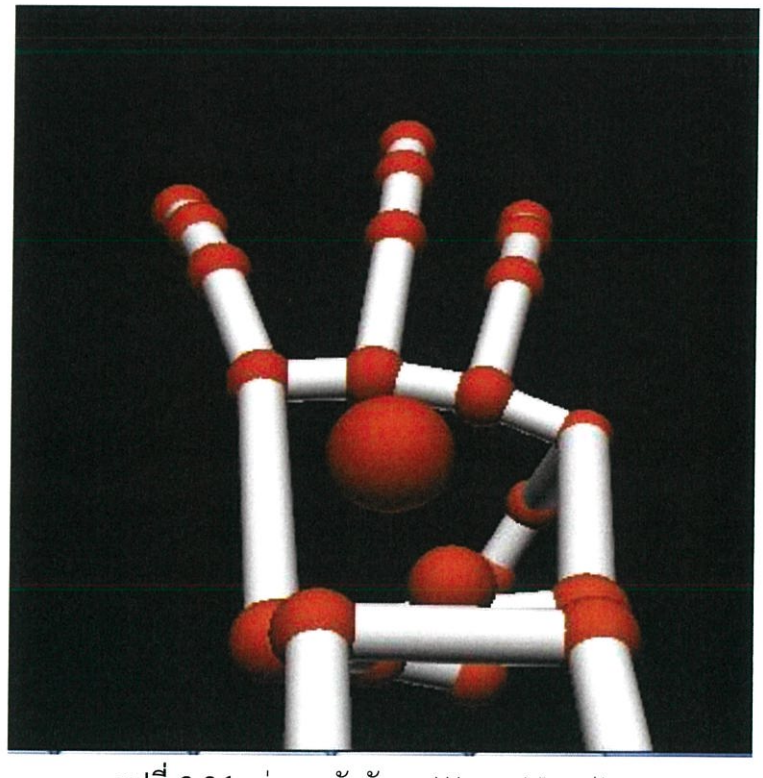
รูปที่ 3.32 ท่าทางตัวอักษร U จาก Visualizer

ตัวอักษร V ชูสองนิ้ว แยกกันตามรูปที่ 3.32



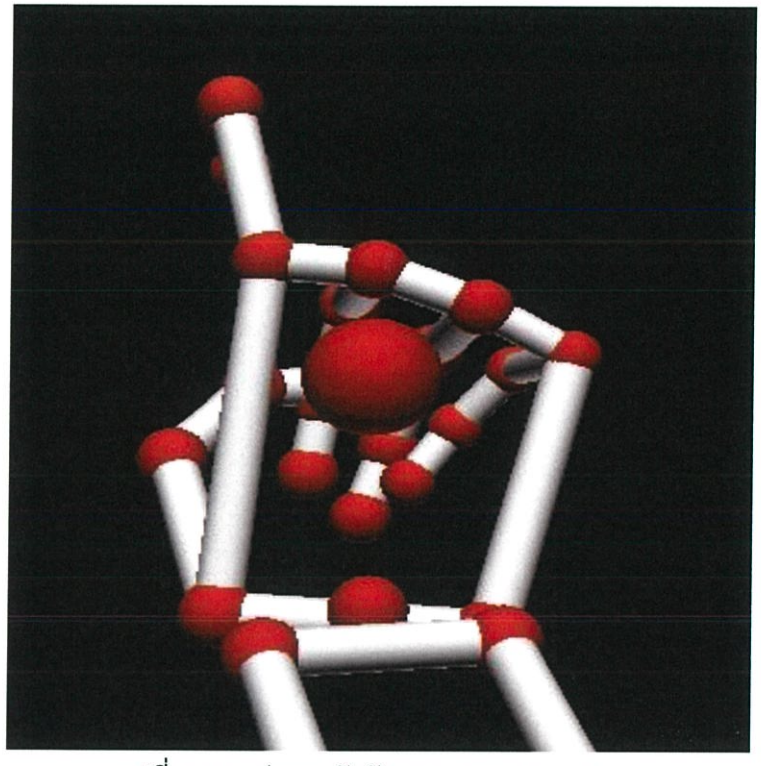
รูปที่ 3.33 ท่าทางตัวอักษร V จาก Visualizer

ตัวอักษร W ชูสามนิ้ว แยกกันตามรูปที่ 3.43



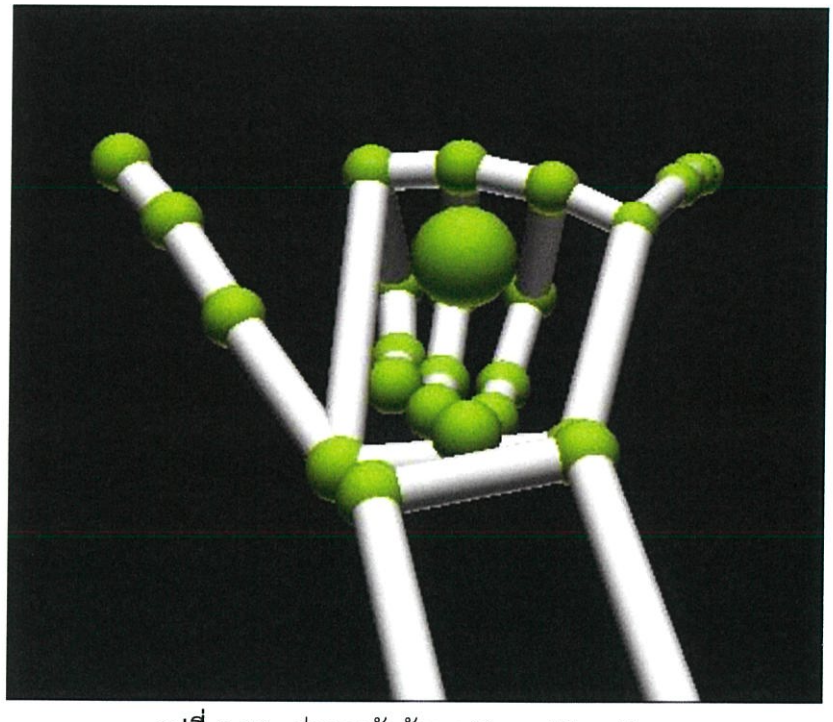
รูปที่ 3.34 ท่าทางตัวอักษร W จาก Visualizer

ตัวอักษร X กำมือ เหยียดนิ้วชี้ตรงแล้วงอข้อต่อแรก



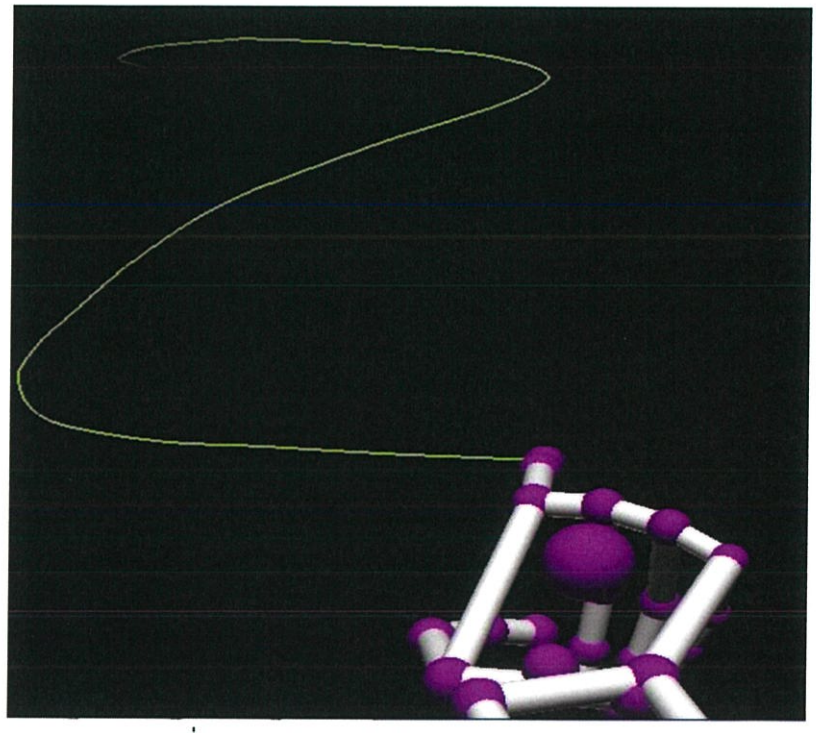
รูปที่ 3.35 ท่าทางตัวอักษร X จาก Visualizer

ตัวอักษร Y กำมือ เหยียดนิ้วหัวแม่มือและนิ้วก้อย



รูปที่ 3.36 ท่าทางตัวอักษร Y จาก Visualizer

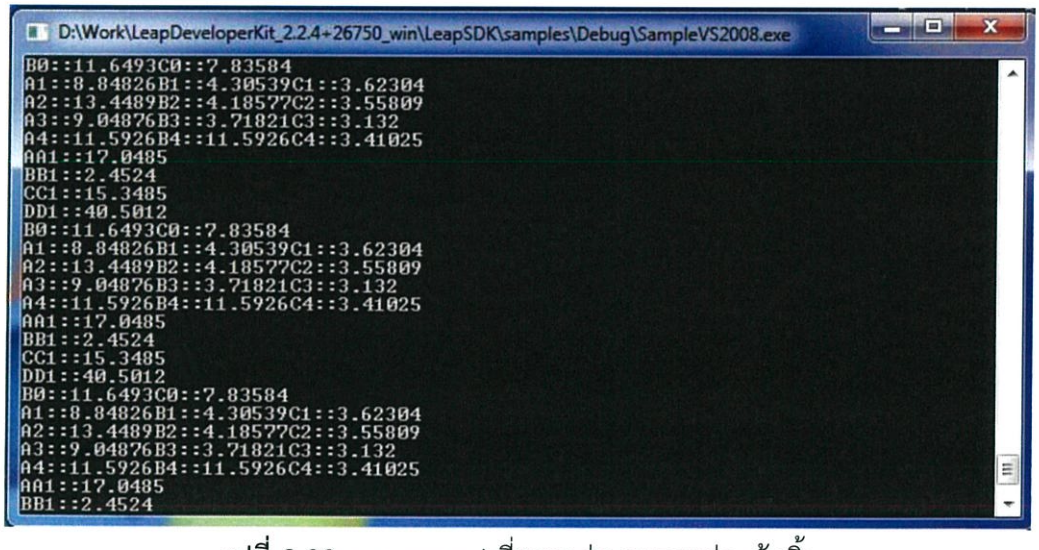
ตัวอักษร Z กำมือ นิ้วชี้เหยียดตรง วาดเป็น ตัว Z



รูปที่ 3.37 ท่าทางตัวอักษร Z จาก Visualizer

3.2.5.3 การ Run program เก็บค่า ตัวอย่างของ Source Code และ ผลลัพธ์ที่ได้

Run program หลังจากทำท่าทางแล้วจะได้ค่ามุมของแต่นิ้ว



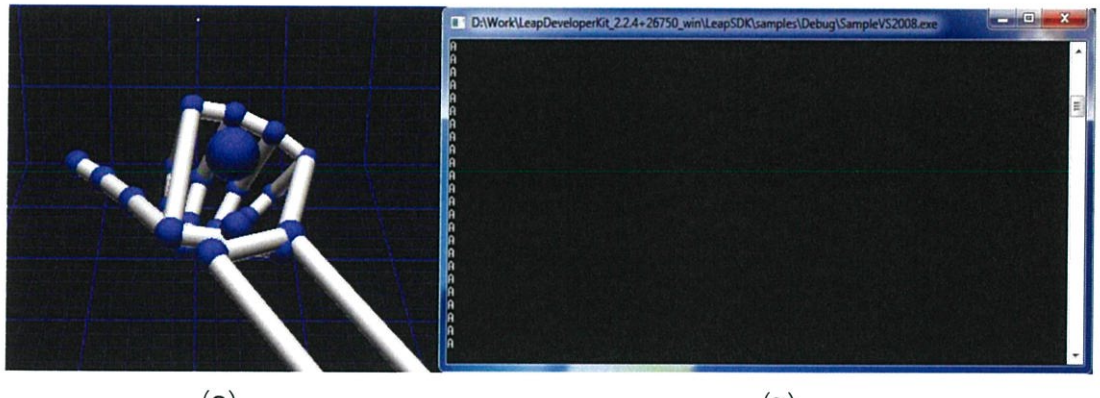
รูปที่ 3.38 command ที่แสดงค่ามุมของแต่ละข้อนิ้ว

เก็บค่าและนำไปเขียน Source code

```
//********* KEEP VALUE OF CHARACTER OF A**********\\
If      (B0 >= 0 && B0 <= 26 && C0 >= 0.5 && C0 <= 29 &&
        A1 >= 49 && A1 <= 84 && B1 >= 54 && B1 <= 90 &&
        C1 >= 25 && C1 <= 45 && A2 >= 60 && A2 <= 90 &&
        B2 >= 70 && B2 <= 90 && C2 >= 28 && C2 <= 45 &&
        A3 >= 59 && A3 <= 90 && B3 >= 69 && B3 <= 90 &&
        C3 >= 25 && C3 <= 46 && A4 >= 59 && A4 <= 90 &&
        B4 >= 69 && B4 <= 89 && C4 >= 26 && C4 <= 46 &&
        AA1 >= 2 && AA1 <= 10 && BB1 >= 2 && BB1 <= 10 &&
        CC1 >= 4 && CC1 <= 15)
{
        std::cout
        << "A"
        << std::endl;
}
```

รูปที่ 3.39 ตัวอย่าง source code

หลังจากเก็บค่าให้มากที่สุดเท่าที่ทำได้ แล้วนำไปใส่ค่าต่ำสุด สูงสุด แยกในแต่ละตัวอักษร ทำการแสดงตัวอักษรอีกครั้งได้ภาพตามรูปที่ 3.39



(ก) (ข)

รูปที่ 3.40 (ก) การแสดงท่าทางเป็นตัวอักษร A
(ข) การแสดงผลลัพธ์เป็นตัวอักษร A

บทที่ 4
การทดลองและผลการทดลอง

ในบทนี้จะอธิบายถึงผลการทดลองที่ได้นำไปสู่การแสดงผลสำหรับการแปลภาษามือในรูปแบบการสะกดคำด้วยนิ้วแบบ American Sign Language (ASL)

4.1 ผลลัพธ์ที่ได้

ทำการทดลองได้ผลลัพธ์ทั้งหมด 23 ตัว จากตัวอักษรทั้งหมด 26 ตัว ตัวที่ไม่สามารถทำได้ ได้แก่ M N T เนื่องจากมีลักษณะท่าทางคล้ายกัน ตัว Leap Motion sensor ไม่สามารถแยกท่าทางของนิ้วที่ซ้อนทับกันโดยละเอียดได้ และทำให้บางท่าทางผิดพลาดจากการจับของ sensor 2 อย่างคือ ไม่แสดงตัวอักษร และ แสดงเป็นตัวอักษรอื่น

4.2 ทดสอบความถูกต้อง

ทดสอบความแม่นยำของท่าทางละ 20 ภาพ กับผู้ทดสอบ 4 คน โดยผลการทดลองสามารถแสดงได้ตามตารางดังต่อไปนี้

ตารางที่ 4.1 ผลการทดลองความแม่นยำในการแปลภาษามือ A – Z ของผู้ทดลองคนที่ 1

ตัวอักษรที่ทำการทดสอบ	จำนวน (ครั้ง)		ตัวอักษรที่ผิดพลาด(ครั้ง)					ร้อยละของความถูกต้อง
	ถูกต้อง	ผิดพลาด	C	D	E	I	S	
A	18	2	1	1	-	-	-	90
B	20	0	-	-	-	-	-	100
C	17	3	-	-	3	-	-	85
D	17	3	2	-	-	-	1	85
E	20	0	-	-	-	-	-	100
F	16	4	2	-	2	-	-	80
G	19	1	1	-	-	-	-	95
H	15	5	2	-	3	-	-	75
I	14	6	-	-	1	-	5	70
J	11	9	-	-	-	9	-	100
K	19	1	1	-	-	-	-	95
L	19	1	-	1	-	-	-	95
M	-	-	-	-	-	-	-	-
N	-	-	-	-	-	-	-	-

O	20	0	-	-	-	-	-	100
P	8	12	3	3	3	-	3	40
Q	20	0	-	-	-	-	-	100
R	18	2	-	2	-	-	-	90
S	10	10	-	-	10	-	-	50
T	-	-	-	-	-	-	-	-
U	15	5	2	3	-	-	-	75
V	18	2	2	-	-	-	-	90
W	14	6	2	-	1	3	-	70
X	17	3	-	3	-	-	-	85
Y	15	5	-	-	-	4	1	75
Z	20	0	-	-	-	-	-	100
ร้อยละความถูกต้องเฉลี่ยของทุกท่าทาง								80.6522

ตารางที่ 4.2 ผลการทดลองความแม่นยำในการแปลภาษามือ A – Z ของผู้ทดลองคนที่ 2

ตัวอักษรที่ทำการทดสอบ	จำนวน (ครั้ง)		ตัวอักษรที่ผิดพลาด(ครั้ง)					ร้อยละของความถูกต้อง
	ถูกต้อง	ผิดพลาด	C	D	E	I	S	
A	16	4	-	-	-	-	4	80
B	17	3	-	-	-	-	-	85
C	3	17	-	-	5	-	2	15
D	13	7	-	-	2	-	-	65
E	20	0	-	-	-	-	-	100
F	15	5	1	-	-	-	-	75
G	14	6	-	-	-	-	-	70
H	12	8	-	-	-	-	-	60
I	0	20	-	-	5	-	6	0
J	0	20	3	1	2	5	2	0
K	8	12	5	-	-	-	-	40
L	0	20	-	-	-	-	-	0
M	-	-	-	-	-	-	-	-
N	-	-	-	-	-	-	-	-
O	18	2	-	-	1	-	1	90
P	0	20	1	-	3	-	-	0

Q	15	5	2	-	-	-	-	75
R	0	20	3	4	2	-	-	0
S	19	1	-	-	1	-	-	95
T	-	-	-	-	-	-	-	-
U	10	10	-	-	-	-	-	50
V	7	13	1	-	5	-	3	35
W	18	2	1	-	-	-	-	90
X	17	3	-	3	-	-	-	85
Y	0	20	-	-	-	5	-	0
Z	15	5	-	5	-	-	-	75
ร้อยละความถูกต้องเฉลี่ยของทุกท่าทาง								51.5217

ตารางที่ 4.3 ผลการทดลองความแม่นยำในการแปลภาษามือ A – Z ของผู้ทดลองคนที่ 3

ตัวอักษรที่ทำการทดสอบ	จำนวน (ครั้ง)		ตัวอักษรที่ผิดพลาด(ครั้ง)					ร้อยละของความถูกต้อง
	ถูกต้อง	ผิดพลาด	C	D	E	I	S	
A	20	0	-	-	-	-	-	100
B	17	3	-	-	-	-	-	85
C	20	0	-	-	-	-	-	100
D	16	4	-	-	-	-	-	80
E	20	0	-	-	-	-	-	100
F	5	15	3	-	5	-	-	25
G	10	10	-	-	-	-	-	50
H	20	0	-	-	-	-	-	100
I	0	20	2	2	2	-	2	0
J	20	0	-	-	-	-	-	100
K	15	5	3	-	-	-	-	75
L	20	0	-	-	-	-	-	100
M	-	-	-	-	-	-	-	-
N	-	-	-	-	-	-	-	-
O	20	0	-	-	-	-	-	100
P	0	20	4	-	5	-	-	0
Q	10	10	-	-	-	-	-	50
R	0	20	2	-	2	-	-	0

S	14	6	2	-	3	-	-	70
T	-	-	-	-	-	-	-	-
U	17	3	-	-	-	-	-	70
V	20	0	-	-	-	-	-	100
W	0	20	1	1	1	-	1	0
X	20	0	-	-	-	-	-	100
Y	20	0	-	-	-	-	-	100
Z	15	5	-	5	-	-	-	75
ร้อยละความถูกต้องเฉลี่ยของทุกท่าทาง								69.3478

ตารางที่ 4.4 ผลการทดลองความแม่นยำในการแปลภาษามือ A – Z ของผู้ทดลองคนที่ 4

ตัวอักษรที่ทำการทดสอบ	จำนวน (ครั้ง)		ตัวอักษรที่ผิดพลาด(ครั้ง)					ร้อยละของความถูกต้อง
	ถูกต้อง	ผิดพลาด	C	D	E	I	S	
A	15	5	-	-	2	-	3	75
B	20	0	-	-	-	-	-	100
C	20	0	-	-	-	-	-	100
D	20	0	-	-	-	-	-	100
E	19	1	1	-	-	-	-	95
F	19	1	1	-	-	-	-	95
G	15	5	2	-	-	-	-	75
H	0	20	5	-	3	-	-	0
I	0	20	-	-	-	-	2	0
J	0	20	-	-	-	5	-	0
K	10	10	-	-	-	-	-	50
L	0	20	5	-	5	-	2	0
M	-	-	-	-	-	-	-	-
N	-	-	-	-	-	-	-	-
O	18	2	-	-	2	-	-	90
P	0	20	3	-	4	-	-	0
Q	15	5	-	-	-	-	-	75
R	0	20	-	5	-	-	-	0
S	15	5	-	-	3	-	-	75
T	-	-	-	-	-	-	-	-

U	0	20	2	5	4	-	-	0
V	20	0	-	-	-	-	-	100
W	0	20	2	-	4	-	-	0
X	10	10	-	10	-	-	-	50
Y	20	0	-	-	-	-	-	100
Z	20	0	-	-	-	-	-	100
ร้อยละความถูกต้องเฉลี่ยของทุกท่าทาง								55.6522

ตารางที่ 4.5 สรุปผลการทดสอบความแม่นยำในการแปลภาษามือ A – Z ของผู้ทดลองทั้งหมด

ตัวอักษรที่ทำการทดสอบ	ร้อยละความถูกต้อง				ค่าเบี่ยงเบนมาตรฐาน (S.D)	ร้อยละความถูกต้องเฉลี่ย
	ผู้ทดลองคนที่ 1	ผู้ทดลองคนที่ 2	ผู้ทดลองคนที่ 3	ผู้ทดลองคนที่ 4		
A	90	80	100	75	4.0041	86.25
B	100	85	85	100	3.1277	92.5
C	85	15	100	100	14.6703	75
D	85	65	80	100	5.2129	82.5
E	100	100	100	95	0.9029	98.75
F	80	75	25	95	10.9718	68.75
G	95	70	50	75	6.6757	72.5
H	75	60	100	0	15.3492	58.75
I	70	0	0	0	12.6405	17.5
J	100	0	100	0	20.8514	50
K	95	40	75	50	8.9685	65
L	95	0	100	0	20.3435	48.75
M	-	-	-	-	-	-
N	-	-	-	-	-	-
O	100	90	100	90	2.0851	95
P	40	0	0	0	7.2232	10
Q	100	75	50	75	7.3721	75
R	90	0	0	0	16.2521	22.5
S	50	95	70	75	6.6757	72.5
T	-	-	-	-	-	-
U	75	50	70	0	12.3689	48.75

V	90	35	100	100	11.2651	81.25
W	70	90	0	0	16.9398	40
X	85	85	100	50	7.6613	80
Y	75	0	100	100	17.0915	68.75
Z	100	75	75	100	5.2129	87.5
ร้อยละความถูกต้องเฉลี่ยของทุกท่าทาง						65.1087

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ปริญญานิพนธ์นี้นำเสนอการแปลภาษามือแบบตัวอักษรภาษาอังกฤษ A – Z หรือ เรียกว่า American Sign Language (ASL) ซึ่งทำการทดลองโดยใช้อุปกรณ์เซนเซอร์สามมิติ Leap Motion ในการจับภาพของมือ โดยเซนเซอร์ตัวนี้จะมีจุดจับที่ข้อต่อของทุกนิ้วแต่ละนิ้วจำนวน 4 ข้อ ใช้เทคนิคการ Dot product หามุมระหว่างข้อนิ้วภายในนิ้วเดียวกัน และ ข้อนิ้วต่างนิ้วกัน โดยกระบวนการจะประกอบไปด้วย กระบวนการรับภาพ กระบวนการประมวลผลข้อมูล และการเปรียบเทียบกับข้อมูลที่ได้ทำการเก็บไว้

5.1 สรุปผลการทดลอง

การทดลองสำหรับปริญญานิพนธ์นี้ เป็นการทดสอบความถูกต้องของ Leap Motion ในการจำแนกท่าทางภาษามือ จากการทดลองค่าร้อยละความถูกต้องเฉลี่ยของทุกท่าทางเท่ากับ 65.1087 ยังมีความแม่นยำแต่ไม่มากพอ เนื่องจากวิธีการขึ้นอยู่กับความชำนาญของผู้ใช้ ผู้ที่ทำท่าทางทดสอบบ่อยจะมีแนวโน้มในการแปลตัวอักษรได้สำเร็จมากกว่าผู้ทดลองที่ทดสอบครั้งแรก ทำให้ข้อมูลที่ได้ยังไม่สามารถนำไปใช้งานได้จริง

5.2 ปัญหาและอุปสรรค

การจับภาพของเซนเซอร์สามมิติงไม่แม่นยำพอที่จะจำแนกบางท่วงท่าได้ แสงที่มากเกินไปจะเป็นปัญหาในการจับของเซนเซอร์สามมิติ

5.3 ข้อเสนอแนะ

หากมีการเก็บข้อมูลที่มากขึ้นจะมีแนวโน้มที่จะแปลตัวอักษรได้สำเร็จมากขึ้น ในตอนนี้ผู้ทดลองไม่สามารถแปลภาษาบางตัวได้เนื่องจากอุปกรณ์ไม่รองรับท่วงท่าที่ซับซ้อน ซึ่งข้อเสนอแนะดังกล่าวควรจะได้รับการพัฒนาต่อไป เพื่อให้ชิ้นงานมีความสมบูรณ์ยิ่งขึ้น เพื่อประโยชน์ในภายภาคหน้า

เอกสารอ้างอิง

[1] M. Mohandes, S. Aliyu and M. Deriche (2014) : "*Arabic Sign Language Recognition using the Leap Motion Controller*" [Online]. IEEE, 2014, pp. 960-965

[2] Wikipedia, "*Trigonometry/Vectors and Dot Products*" : Wikibook, 2011.

[3] พ.ญ. เพ็ญมาศ ธรรมศรัณยู. "*Noise-Induced Hearing Loss*", วันที่ 24 พฤษภาคม 2554.

[4] ไกรศร ตั้งโอภากุล. คู่มือเรียนเขียนโปรแกรมภาษา C, นนทบุรี : ไอซีดี. 2554.

[5] "*Leap Motion*" [Online]. เข้าถึงได้จาก : http://pantip.com/topic/30833642

[6] Leap team, "*Leap Motion*" [Online].
เข้าถึงได้จาก : http://www.leapmotion.com/

[7] Yodiagrace, "*ASL*" [Online]. เข้าถึงได้จาก :
http://yodiagrace.wordpress.com/2012/05/06/มารู้จัก-asl-กันดีกว่า

[8] เว็บไซต์ MED-EL international ประเทศไทย. "*โครงสร้างของหู*" [Online].
เข้าถึงได้จาก : http://www.medel.com/th/anatomy-of-the-ear

[9] ฝ่ายบริการสนับสนุนนักศึกษาพิการเรียนร่วม (Disability Support Services หรือ DSS) หลักสูตรการศึกษาพิเศษ คณะครุศาสตร์ มหาวิทยาลัยราชภัฏสวนดุสิต. "*ภาษามือ*" [Online].
เข้าถึงได้จาก : http://www.dssdusit.com/aural/sign/

[10] ศูนย์การศึกษาพิเศษประจำจังหวัดชัยนาท. "*บุคคลที่มีความบกพร่องทางการได้ยิน*" [Online].
เข้าถึงได้จาก : http://www.chainatspecial.com/wall/data/handicapped%20hear.pdf/

ภาคผนวก ก

```
//*****************************
PROGRAM FOR SIGN LANGUAGE By
MEASURING ANGLE OF THE BENDIND
FINGER ******************************//

#include <iostream>

#include <string.h>

#include "Leap.h"

#include <math.h>

#include <string>

#include <windows.h>

using namespace Leap;

class SampleListener : public Listener {

public:

        virtual void onInit(const
Controller&);

        virtual void onConnect(const
Controller&);

        virtual void onDisconnect(const
Controller&);

        virtual void onExit(const
Controller&);

        virtual void onFrame(const
Controller&);

        virtual void
onFocusGained(const Controller&);

        virtual void onFocusLost(const
Controller&);

        virtual void
onDeviceChange(const Controller&);

        virtual void
onServiceConnect(const Controller&);

        virtual void
onServiceDisconnect(const
Controller&);

private:

};


const std::string fingerNames[] = {
"Thumb", "Index", "Middle", "Ring",
"Pinky" };

const std::string boneNames[] = {
"Metacarpal", "Proximal", "Middle",
"Distal" };

const std::string stateNames[] = {
"STATE_INVALID", "STATE_START",
"STATE_UPDATE", "STATE_END" };

void SampleListener::onInit(const
Controller& controller) {

}

void SampleListener::onConnect(const
Controller& controller) {

        controller.enableGesture(Gestu
re::TYPE_CIRCLE);

        controller.enableGesture(Gestu
re::TYPE_KEY_TAP);
```

```
        controller.enableGesture(Gestu
re::TYPE_SCREEN_TAP);

        controller.enableGesture(Gestu
re::TYPE_SWIPE);

}

void
SampleListener::onDisconnect(const
Controller& controller) {

}

void SampleListener::onExit(const
Controller& controller) {

}

void SampleListener::onFrame(const
Controller& controller) {

        const Frame frame =
controller.frame();

        HandList hands =
frame.hands();

        char s = 1;

        for (HandList::const_iterator hl
= hands.begin(); hl != hands.end();
++hl) {

                const Hand hand = *hl;

                const Vector normal =
hand.palmNormal();

                const Vector direction =
hand.direction();

                Arm arm = hand.arm();

                const FingerList fingers
= hand.fingers();

                for
(FingerList::const_iterator fl =
fingers.begin(); fl != fingers.end(); ++fl) {

                        const Finger
finger = *fl;

                        const
GestureList gestures = frame.gestures();

                        const ToolList
tools = frame.tools();

                        const
PointableList inFrame =
frame.pointables();
        ////////////////////////////////
////////////////////////////////////////
//////////////////

                        Finger::Type
fingerType = finger.type();

                        Bone
metacarpal =
finger.bone(Bone::Type::TYPE_METACA
RPAL);

                        Bone
intermediate =
finger.bone(Bone::Type::TYPE_INTERME
DIATE);

                        Bone proximal =
finger.bone(Bone::Type::TYPE_PROXIMA
L);
```

```
                    Bone distal =
finger.bone(Bone::Type::TYPE_DISTAL);

        //******************** IF sensor
capture hand*****************//
                    if
(!frame.hands().isEmpty())
                    {
                        float B0;
                        float C0;
                        float A1;
                        float B1;
                        float C1;
                        float A2;
                        float B2;
                        float C2;
                        float A3;
                        float B3;
                        float C3;
                        float A4;
                        float B4;
                        float C4;
                        float
AA1;
                        float
BB1;
                        float
CC1;
                        float
DD1;
                        float EE1;
                        float
stop;
                        int K = 0;
                        Vector g;
                        Vector p;

        //*************** THUMB FINGER
????????????????

        Leap::Vector T1 =
hand.fingers()[Leap::Finger::Type::TYPE_
THUMB].bone(Leap::Bone::Type::TYPE_
PROXIMAL).direction();

        Leap::Vector T2 =
hand.fingers()[Leap::Finger::Type::TYPE_
THUMB].bone(Leap::Bone::Type::TYPE_I
NTERMEDIATE).direction();

        Leap::Vector T3 =
hand.fingers()[Leap::Finger::Type::TYPE_
THUMB].bone(Leap::Bone::Type::TYPE_
DISTAL).direction();
                        /////To
measure Angle of the bending finger
```

```
between proximal bone with
Intermediate bone

                                float T00
= T1.dot(T2);

                                float
tproxx = T1[0];

                                float
tproxy = T1[1];

                                float
tproxz = T1[2];

                                float
tinterx = T2[0];

                                float
tintery = T2[1];

                                float
tinterz = T2[2];

                                float PT1
= (pow(tproxx, 2) + pow(tproxy, 2) +
pow(tproxz, 2));

                                float PT2
= (pow(tinterx, 2) + pow(tintery, 2) +
pow(tinterz, 2));

                                float TQ1
= sqrt(PT1);

                                float TQ2
= sqrt(PT2);

                                float TAA
= (TQ1*TQ2);

                                float TPI
= acos(T00 / TAA);

                                float
ATPI = TPI * (180 / PI);

                                B0 =
ATPI;

                                //To
measure angle of the bending finger
between intermediate phalanges bone
with distal phalanges bone/////////

                                float T01
= T2.dot(T3);

                                float
ainterx = T2[0];

                                float
aintery = T2[1];

                                float
ainterz = T2[2];

                                float
adisx = T3[0];

                                float
adisy = T3[1];

                                float
adisz = T3[2];

                                float PT3
= (pow(ainterx, 2) + pow(aintery, 2) +
pow(ainterz, 2));

                                float PT4
= (pow(adisx, 2) + pow(adisy, 2) +
pow(adisz, 2));
```

```
                                        float TQ3
= sqrt(PT3);
                                        float TQ4
= sqrt(PT4);
                                        float TAB
= (TQ3*TQ4);
                                        float TID
= acos(T01 / TAB);
                                        float
ATID = TID* (180 / PI);
                                        C0 =
ATID;

        /*std::cout

        << "B0::" << B0

        << "C0::" << C0

        << std::endl;*/

        //*************** INDEX FINGER
????????????????????

        Leap::Vector I1 =
hand.fingers()[Leap::Finger::Type::TYPE_
INDEX].bone(Leap::Bone::Type::TYPE_M
ETACARPAL).direction();

        Leap::Vector I2 =
hand.fingers()[Leap::Finger::Type::TYPE_
INDEX].bone(Leap::Bone::Type::TYPE_P
ROXIMAL).direction();

        Leap::Vector I3 =
hand.fingers()[Leap::Finger::Type::TYPE_
INDEX].bone(Leap::Bone::Type::TYPE_IN
TERMEDIATE).direction();

        Leap::Vector I4 =
hand.fingers()[Leap::Finger::Type::TYPE_
INDEX].bone(Leap::Bone::Type::TYPE_DI
STAL).direction();

        Sleep(25);

                                        /////To
measure Angle of the bending finger
between  Metacarpal bone  with
Proximal  bone
                                        float I00
= I1.dot(I2);
                                        float
imetx = I1[0];
                                        float
imety = I1[1];
                                        float
imetz = I1[2];
                                        float
iproxx = I2[0];
                                        float
iproxy = I2[1];
```

```
                                        float
iproxz = I2[2];

                                        float PI1
= (pow(imetx, 2) + pow(imety, 2) +
pow(imetz, 2));

                                        float PI2
= (pow(iproxx, 2) + pow(iproxy, 2) +
pow(iproxz, 2));

                                        float IQ1
= sqrt(PI1);

                                        float IQ2
= sqrt(PI2);

                                        float IAA
= (IQ1*IQ2);

                                        float IMP
= acos(I00 / IAA);

                                        float
AIMP = IMP* (180 / PI);

                                        A1 =
AIMP

                                        /////To
measure Angle of the bending finger
between  proximal bone  with
Intermediate  bone

                                        float I01
= I2.dot(I3);

                                        float
iiproxx = I2[0];

                                        float
iiproxy = I2[1];

                                        float
iiproxz = I2[2];

                                        float
iinterx = I3[0];

                                        float
iintery = I3[1];

                                        float
iinterz = I3[2];

                                        float PI3
= (pow(iiproxx, 2) + pow(iiproxy, 2) +
pow(iiproxz, 2));

                                        float PI4
= (pow(iinterx, 2) + pow(iintery, 2) +
pow(iinterz, 2));

                                        float IQ3
= sqrt(PI3);

                                        float IQ4
= sqrt(PI4);

                                        float IAB
= (IQ3*IQ4);

                                        float IPI
= acos(I01 / IAB);

                                        float AIPI
= IPI* (180 / PI);

                                        B1 = AIPI;

                                        //To
measure angle of the bending finger
between intermediate phalanges bone
with distal phalanges bone/////////
```

```
                                float I02
= I3.dot(I4);

                                float
iiinterx = I3[0];

                                float
iiintery = I3[1];

                                float
iiinterz = I3[2];

                                float
idisx = I4[0];

                                float
idisy = I4[1];

                                float
idisz = I4[2];

                                float PI5
= (pow(iiinterx, 2) + pow(iiintery, 2) +
pow(iiinterz, 2));

                                float PI6
= (pow(idisx, 2) + pow(idisy, 2) +
pow(idisz, 2));

                                float IQ5
= sqrt(PI5);

                                float IQ6
= sqrt(PI6);

                                float IAC
= (IQ5*IQ6);

                                float IID
= acos(I02 / IAC);

                                float AIID
= IID* (180 / PI);

                                C1 =
AIID;

        /*std::cout

        << "A1::" << A1

        << "B1::" << B1

        << "C1::" << C1

        << std::endl;*/

        //*************** MIDDEL FINGER
????????????????

        Leap::Vector M1 =
hand.fingers()[Leap::Finger::Type::TYPE_
MIDDLE].bone(Leap::Bone::Type::TYPE_
METACARPAL).direction();

        Leap::Vector M2 =
hand.fingers()[Leap::Finger::Type::TYPE_
MIDDLE].bone(Leap::Bone::Type::TYPE_
PROXIMAL).direction();

        Leap::Vector M3 =
hand.fingers()[Leap::Finger::Type::TYPE_
MIDDLE].bone(Leap::Bone::Type::TYPE_I
NTERMEDIATE).direction();
```

```
        Leap::Vector M4 =
hand.fingers()[Leap::Finger::Type::TYPE_
MIDDLE].bone(Leap::Bone::Type::TYPE_
DISTAL).direction();

                                    /////To
measure Angle of the bending finger
between  Metacarpal bone  with
Proximal  bone

                                    float
M00 = M1.dot(M2);

                                    float
mmetx = M1[0];

                                    float
mmety = M1[1];

                                    float
mmetz = M1[2];

                                    float
mproxx = M2[0];

                                    float
mproxy = M2[1];

                                    float
mproxz = M2[2];

                                    float
PM1 = (pow(mmetx, 2) + pow(mmety,
2) + pow(mmetz, 2));

                                    float
PM2 = (pow(mproxx, 2) + pow(mproxy,
2) + pow(mproxz, 2));

                                    float
MQ1 = sqrt(PM1);

                                    float
MQ2 = sqrt(PM2);

                                    float
MAA = (MQ1*MQ2);

                                    float
MMP = acos(M00 / MAA);

                                    float
AMMP = MMP* (180 / PI);

                                    A2 =
AMMP;

//////////////////////////To measure
Angle of the bending finger  between
proximal bone  with  Intermediate
bone

                                    float
M01 = M2.dot(M3);

                                    float
mmproxx = M2[0];

                                    float
mmproxy = M2[1];

                                    float
mmproxz = M2[2];

                                    float
minterx = M3[0];

                                    float
mintery = M3[1];

                                    float
minterz = M3[2];
```

```
                                        float
PM3 = (pow(mmproxx, 2) +
pow(mmproxy, 2) + pow(mmproxz, 2));

                                        float
PM4 = (pow(minterx, 2) + pow(mintery,
2) + pow(minterz, 2));

                                        float
MQ3 = sqrt(PM3);

                                        float
MQ4 = sqrt(PM4);

                                        float
MAB = (MQ3*MQ4);

                                        float MPI
= acos(M01 / MAB);

                                        float
AMPI = MPI* (180 / PI);

                                        B2 =
AMPI;

//To measure angle of the bending
finger between intermediate phalanges
bone with distal phalanges
bone/////////

                                        float
M02 = M3.dot(M4);

                                        float
mminterx = M3[0];

                                        float
mmintery = M3[1];

                                        float
mminterz = M3[2];

                                        float
mdisx = M4[0];

                                        float
mdisy = M4[1];

                                        float
mdisz = M4[2];

                                        float
PM5 = (pow(mminterx, 2) +
pow(mmintery, 2) + pow(mminterz, 2));

                                        float
PM6 = (pow(mdisx, 2) + pow(mdisy, 2)
+ pow(mdisz, 2));

                                        float
MQ5 = sqrt(PM5);

                                        float
MQ6 = sqrt(PM6);

                                        float
MAC = (MQ5*MQ6);

                                        float MID
= acos(M02 / MAC);

                                        float
AMID = MID* (180 / PI);

                                        C2 =
AMID;

        /*std::cout

        << "A2::" << A2

        << "B2::" << B2
```

```
        << "C2::" << C2

        << std::endl;*/

        //*************** RING FINGER
????????????????????

        Leap::Vector R1 =
hand.fingers()[Leap::Finger::Type::TYPE_
RING].bone(Leap::Bone::Type::TYPE_ME
TACARPAL).direction();

        Leap::Vector R2 =
hand.fingers()[Leap::Finger::Type::TYPE_
RING].bone(Leap::Bone::Type::TYPE_PR
OXIMAL).direction();

        Leap::Vector R3 =
hand.fingers()[Leap::Finger::Type::TYPE_
RING].bone(Leap::Bone::Type::TYPE_INT
ERMEDIATE).direction();

        Leap::Vector R4 =
hand.fingers()[Leap::Finger::Type::TYPE_
RING].bone(Leap::Bone::Type::TYPE_DIS
TAL).direction();

                                ////To
measure Angle of the bending finger
between  Metacarpal bone  with
Proximal  bone

                                float R00
 = R1.dot(R2);

                                float
rmetx = R1[0];

                                float
rmety = R1[1];

                                float
rmetz = R1[2];

                                float
rproxx = R2[0];

                                float
rproxy = R2[1];

                                float
rproxz = R2[2];

                                float PR1
= (pow(rmetx, 2) + pow(rmety, 2) +
pow(rmetz, 2));

                                float PR2
= (pow(rproxx, 2) + pow(rproxy, 2) +
pow(rproxz, 2));

                                float RQ1
= sqrt(PR1);

                                float RQ2
= sqrt(PR2);

                                float RAA
= RQ1 * RQ2;

                                float
RMP = acos(R00 / RAA);

                                float
ARMP = RMP * (180 / PI);
```

```
A3 = ARMP;

/////To measure Angle of the bending finger between proximal bone with Intermediate bone

float R01 = R2.dot(R3);

float rrproxx = R2[0];

float rrproxy = R2[1];

float rrproxz = R2[2];

float rinterx = R3[0];

float rintery = R3[1];

float rinterz = R3[2];

float PR3 = (pow(rrproxx, 2) + pow(rrproxy, 2) + pow(rrproxz, 2));

float PR4 = (pow(rinterx, 2) + pow(rintery, 2) + pow(rinterz, 2));

float RQ3 = sqrt(PR3);

float RQ4 = sqrt(PR4);

float RAB = RQ3 * RQ4;

float RPI = acos(R01 / RAB);

float ARPI = RPI * (180 / PI);

B3 = ARPI;

//To measure angle of the bending finger between intermediate phalanges bone with distal phalanges bone/////////

float R02 = R3.dot(R4);

float rrinterx = R3[0];

float rrintery = R3[1];

float rrinterz = R3[2];

float rdisx = R4[0];

float rdisy = R4[1];

float rdisz = R4[2];

float PR5 = (pow(rrinterx, 2) + pow(rrintery, 2) + pow(rrinterz, 2));
```

```
                                float PR6
= (pow(rdisx, 2) + pow(rdisy, 2) +
pow(rdisz, 2));

                                float RQ5
= sqrt(PR5);

                                float RQ6
= sqrt(PR6);

                                float RAC
= (RQ5*RQ6);

                                float RID
= acos(R02 / RAC);

                                float
ARID = RID* (180 / PI);

                                C3 =
ARID;

        /*std::cout

        << "A3::" << A3

        << "B3::" << B3

        << "C3::" << C3

        << std::endl;*/


        //*************** PINKY FINGER
????????????????????


        Leap::Vector P1 =
hand.fingers()[Leap::Finger::Type::TYPE_
PINKY].bone(Leap::Bone::Type::TYPE_M
ETACARPAL).direction();


        Leap::Vector P2 =
hand.fingers()[Leap::Finger::Type::TYPE_
PINKY].bone(Leap::Bone::Type::TYPE_P
ROXIMAL).direction();


        Leap::Vector P3 =
hand.fingers()[Leap::Finger::Type::TYPE_
PINKY].bone(Leap::Bone::Type::TYPE_IN
TERMEDIATE).direction();


        Leap::Vector P4 =
hand.fingers()[Leap::Finger::Type::TYPE_
PINKY].bone(Leap::Bone::Type::TYPE_DI
STAL).direction();

                                ////To
measure Angle of the bending finger
between  Metacarpal bone  with
Proximal  bone

                                float P00
= P1.dot(P2);

                                float
pmetx = P1[0];

                                float
pmety = P1[1];

                                float
pmetz = P1[2];
```

```
                                        float
pproxx = P2[0];

                                        float
pproxy = P2[1];

                                        float
pproxz = P2[2];

                                        float PP1
= (pow(pmetx, 2) + pow(pmety, 2) +
pow(pmetz, 2));

                                        float PP2
= (pow(pproxx, 2) + pow(pproxy, 2) +
pow(pproxz, 2));

                                        float PQ1
= sqrt(PP1);

                                        float PQ2
= sqrt(PP2);

                                        float PAA
= (PQ1*PQ2);

                                        float
PMP = acos(P00 / PAA);

                                        float
APMP = PMP* (180 / PI);

                                        A4 =
APMP;

                                        /////To
measure Angle of the bending finger
between  proximal bone  with
Intermediate  bone

                                        float P01
= P2.dot(P3);
```

```
                                        float
ppproxx = P1[0];

                                        float
ppproxy = P1[1];

                                        float
ppproxz = P1[2];

                                        float
pinterx = P2[0];

                                        float
pintery = P2[1];

                                        float
pinterz = P2[2];

                                        float PP3
= (pow(ppproxx, 2) + pow(ppproxy, 2)
+ pow(ppproxz, 2));

                                        float PP4
= (pow(pinterx, 2) + pow(pintery, 2) +
pow(pinterz, 2));

                                        float PQ3
= sqrt(PP3);

                                        float PQ4
= sqrt(PP4);

                                        float PAB
= (PQ3*PQ4);

                                        float PPI
= acos(P00 / PAB);

                                        float
APPI = PPI* (180 / PI);

                                        B4 =
APPI;
```

```
//To measure angle of the bending finger between intermediate phalanges bone with distal phalanges bone/////////

float P02 = P3.dot(P4);

float ppinterx = P3[0];

float ppintery = P3[1];

float ppinterz = P3[2];

float pdisx = P4[0];

float pdisy = P4[1];

float pdisz = P4[2];

float PP5 = (pow(ppinterx, 2) + pow(ppintery, 2) + pow(ppinterz, 2));

float PP6 = (pow(pdisx, 2) + pow(pdisy, 2) + pow(pdisz, 2));

float PQ5 = sqrt(PP5);

float PQ6 = sqrt(PP6);

float PAC = (PQ5*PQ6);

float PID = acos(P02 / PAC);

float APID = PID* (180 / PI);

C4 = APID;

/*std::cout

<< "A4::" << A4

<< "B4::" << B4

<< "C4::" << C4

<< std::endl;*/

//To measure angle between index finger and middle finger proximal joint//

float IM00 = I2.dot(M2);

float iiiproxx = I2[0];

float iiiproxy = I2[1];

float iiiproxz = I2[2];

float mmmproxx = M2[0];
```

```
                                    float
mmmproxy = M2[1];

                                    float
mmmproxz = M2[2];

                                    float
PIM1 = (pow(iiiproxx, 2) + pow(iiiproxy,
2) + pow(iiiproxz, 2));

                                    float
PIM2 = (pow(mmmproxx, 2) +
pow(mmmproxy, 2) + pow(mmmproxz,
2));

                                    float
IMQ1 = sqrt(PIM1);

                                    float
IMQ2 = sqrt(PIM2);

                                    float
IMAA = (IMQ1*IMQ2);

                                    float
IMPP = acos(IM00 / PAC);

                                    float
AIMPP = IMPP* (180 / PI);

                                    AA1 =
AIMPP;


        /*std::cout


        << "AA1::" << AA1


        << std::endl;*/
```

```
                                    //To
measure angle between middle finger
and ring finger proximal joint//

                                    float
MR00 = M2.dot(R2);

                                    float
mmmmproxx = M2[0];

                                    float
mmmmproxy = M2[1];

                                    float
mmmmproxz = M2[2];

                                    float
rrrproxx = R2[0];

                                    float
rrrproxy = R2[1];

                                    float
rrrproxz = R2[2];

                                    float
PMR1 = (pow(mmmmproxx, 2) +
pow(mmmmproxy, 2) +
pow(mmmmproxz, 2));

                                    float
PMR2 = (pow(rrrproxx, 2) +
pow(rrrproxy, 2) + pow(rrrproxz, 2));

                                    float
MRQ1 = sqrt(PMR1);

                                    float
MRQ2 = sqrt(PMR2);

                                    float
MRAA = (MRQ1*MRQ2);
```

```
                                    float
MRPP = acos(MR00 / PAC);

                                    float
AMRPP = MRPP* (180 / PI);

                                    BB1 =
AMRPP;

        /*std::cout

        << "BB1::" << BB1

        << std::endl;*/


                                    //To
measure angle between ring finger and
pinky finger proximal joint//

                                    float
RP00 = R2.dot(P2);

                                    float
rrrrproxx = R2[0];

                                    float
rrrrproxy = R2[1];

                                    float
rrrrproxz = R2[2];

                                    float
ppppproxx = P2[0];

                                    float
pppproxy = P2[1];

                                    float
ppppproxz = P2[2];


                                    float
PRP1 = (pow(rrrrproxx, 2) +
pow(rrrrproxy, 2) + pow(rrrrproxz, 2));

                                    float
PRP2 = (pow(ppppproxx, 2) +
pow(pppproxy, 2) + pow(ppppproxz, 2));

                                    float
RPQ1 = sqrt(PRP1);

                                    float
RPQ2 = sqrt(PRP2);

                                    float
RPAA = (RPQ1*RPQ2);

                                    float
RPPP = acos(RP00 / PAC);

                                    float
ARPPP = RPPP* (180 / PI);

                                    CC1 =
ARPPP;

        /*std::cout

        << "CC1::" << CC1

        << std::endl;*/


                                    //To
measure angle between thumb finger
and index finger distal joint//

                                    float TI00
= T3.dot(I4);
```

```
                                        float
tttdisx = T3[0];
                                        float
tttdisy = T3[1];
                                        float
tttdisz = T3[2];
                                        float
iiidisx = I4[0];
                                        float
iiidisy = I4[1];
                                        float
iiidisz = I4[2];
                                        float
TIDP1 = (pow(tttdisx, 2) + pow(tttdisy,
2) + pow(tttdisz, 2));
                                        float
TIDP2 = (pow(iiidisx, 2) + pow(iiidisy, 2)
+ pow(iiidisz, 2));
                                        float
TIDPQ1 = sqrt(TIDP1);
                                        float
TIDPQ2 = sqrt(TIDP2);
                                        float
TIDAA = (TIDPQ1*TIDPQ2);
                                        float
TIDPP = acos(TI00 / PAC);
                                        float
ATIDPP = TIDPP* (180 / PI);
                                        DD1 =
ATIDPP;

        /*std::cout

        << "DD1::" << DD1

        << std::endl;*/

                                        //To
measure angle between thumb finger
and middle finger distal joint//
                                        float
TM00 = T3.dot(M4);
                                        float
ttttdisx = T3[0];
                                        float
ttttdisy = T3[1];
                                        float
ttttdisz = T3[2];
                                        float
mmmmdisx = M4[0];
                                        float
mmmmdisy = M4[1];
                                        float
mmmmdisz = M4[2];
                                        float
TMDP1 = (pow(ttttdisx, 2) +
pow(ttttdisy, 2) + pow(ttttdisz, 2));
                                        float
TMDP2 = (pow(mmmmdisx, 2) +
pow(mmmmdisy, 2) +
pow(mmmmdisz, 2));
```

```
                                float
TMDPQ1 = sqrt(TMDP1);
                                float
TMDPQ2 = sqrt(TMDP2);
                                float
TMDAA = (TMDPQ1*TMDPQ2);
                                float
TMDPP = acos(TM00 / PAC);
                                float
ATMDPP = TMDPP* (180 / PI);
                                EE1 =
ATMDPP;

        /*std::cout
                                << "EE1::"
<< EE1
                                <<
std::endl;*/
                                float AA
= hand.pinchStrength();
                                float BB
= AA * 100;
                                stop =
BB;
                                std::cout
                                <<
"stop::" << stop
                                <<
std::endl;

        //*****************************
************** FINISH FOR THE BENDING
FINGER
*****************************************
*****************************//

                                // Get
gestures
                                const
GestureList gestures = frame.gestures();
                                for (int g
= 0; g < gestures.count(); ++g) {

        Gesture gesture = gestures[g];

        switch (gesture.type()) {

        case Gesture::TYPE_CIRCLE:
                                        {

        CircleGesture circle = gesture;

        K = 1;

        if (circle.state() !=
Gesture::STATE_START &&

                B0 >= 1 && B0 <= 55
&&
```

```
                C0 >= 2 && C0 <= 55
&&
                A1 >= 60 && A1 <= 90
&&
                B1 >= 50 && B1 <= 95
&&
                C1 >= 15 && C1 <= 55
&&
                A2 >= 65 && A2 <= 90
&&
                B2 >= 50 && B2 <= 90
&&
                C2 >= 10 && C2 <= 55
&&
                A3 >= 35 && A3 <= 89
&&
                B3 >= 40 && B3 <= 89
&&
                C3 >= 15 && C3 <= 45
&&
                A4 >= 0 && A4 <= 40
&&
                B4 >= 2 && B4 <= 29
&&
                C4 >= 1 && C4 <= 25)
            {
                std::cout
                    << "J"
                    << std::endl;
            }
            break;
        }
        case Gesture::TYPE_SWIPE:
        {
            SwipeGesture swipe = gesture;
            if (swipe.state() !=
Gesture::STATE_START&&
```

```
B0 >= 5 && B0 <= 55
&&
C0 >= 20 && C0 <= 65
&&
A1 >= 0 && A1 <= 30
&&
B1 >= 0 && B1 <= 35
&&
C1 >= 0 && C1 <= 35
&&
A2 >= 45 && A2 <= 98
&&
B2 >= 45 && B2 <= 90
&&
C2 >= 10 && C2 <= 55
&&
A3 >= 35 && A3 <= 89
&&
B3 >= 40 && B3 <= 89
&&
C3 >= 15 && C3 <= 65
&&
A4 >= 30 && A4 <= 120
&&
B4 >= 45 && B4 <= 99
&&
C4 >= 15 && C4 <= 65)
{
std::cout
<< "Z"
<< std::endl;
}
break;
}
case Gesture::TYPE_KEY_TAP:
{
KeyTapGesture tap = gesture;
```

```
                                                                    if
        break;

                                    }               (B0 >= 0 && B0 <= 26 &&

        case                                        C0 >= 0.5 && C0 <= 29 &&
Gesture::TYPE_SCREEN_TAP:

                                    {               A1 >= 49 && A1 <= 84 &&

        ScreenTapGesture screentap =                B1 >= 54 && B1 <= 90 &&
gesture;

                                                    C1 >= 25 && C1 <= 45 &&
        break;

                                    }               A2 >= 60 && A2 <= 90 &&

        default:                                    B2 >= 70 && B2 <= 90 &&

        std::cout << std::string(2, ' ') <<         C2 >= 28 && C2 <= 45 &&
"Unknown gesture type." << std::endl;

                                                    A3 >= 59 && A3 <= 90 &&
        ////////////////////

                                                    B3 >= 69 && B3 <= 90 &&
        /////////////////

                                                    C3 >= 25 && C3 <= 46 &&
        break;

                                    }               A4 >= 59 && A4 <= 90 &&

                                }

                        //********* KEEP            B4 >= 69 && B4 <= 89 &&
VALUE OF CHARACTER OF
A**********\\                                       C4 >= 26 && C4 <= 46 &&
```

```
        AA1 >= 2 && AA1 <= 10 &&

        BB1 >= 2 && BB1 <= 10 &&

        CC1 >= 4 && CC1 <= 15)

        {

        std::cout

        << "A"

        << std::endl;

        }

                        //********* KEEP
VALUE OF CHARACTER OF
B**********\\

                                if (B0 >=
29 && B0 <= 44 &&

        C0 >= 3 && C0 <= 69 &&

        A1 >= 5 && A1 <= 46 &&

        B1 >= 2 && B1 <= 47 &&

        C1 >= 1.6 && C1 <= 27 &&

        A2 >= 2.5 && A2 <= 36 &&

        B2 >= 1.6 && B2 <= 28 &&

        C2 >= 1.4 && C2 <= 22 &&

        A3 >= 6.8 && A3 <= 30 &&

        B3 >= 1.5 && B3 <= 20 &&

        C3 >= 1.2 && C3 <= 14 &&

        A4 >= 4 && A4 <= 30 &&

        B4 >= 2 && B4 <= 30 &&

        C4 >= 1.4 && C4 <= 11 &&

        AA1 >= 7 && AA1 <= 20 &&

        BB1 >= 1 && BB1 <= 10 &&

        CC1 >= 4 && CC1 <= 15)
                                {

        std::cout

        << "B"
```

```
<< std::endl;
                }
        //********* KEEP
VALUE OF CHARACTER OF
C**********\\
        if (B0 >=
0.1 && B0 <= 40 &&
        C0 >= 1
&& C0 <= 30 &&
        A1 >= 3
&& A1 <= 60 &&
        B1 >= 30
&& B1 <= 80 &&
        C1 >= 20
&& C1 <= 60 &&
        A2 >= 5
&& A2 <= 60 &&
        B2 >= 30
&& B2 <= 80 &&
        C2 >= 20
&& C2 <= 65 &&
        A3 >= 15
&& A3 <= 60 &&
        B3 >= 5
&& B3 <= 80 &&
        C3 >= 4
&& C3 <= 65 &&
        A4 >= 5
&& A4 <= 80 &&
        B4 >= 5
&& B4 <= 80 &&
        C4 >= 4
&& C4 <= 60 &&
        AA1 >=
0.1 && AA1 <= 20 &&
        BB1 >= 2
&& BB1 <= 40 &&
        CC1 >= 2
&& CC1 <= 30 )
            {
    std::cout
    << "C"
    << std::endl;
            }
        //********* KEEP
VALUE OF CHARACTER OF
D**********\\
        if (B0 >=
20 && B0 <= 42 &&
    C0 >= 8.5 && C0 <= 55 &&
    A1 >= 1.8 && A1 <= 25 &&
    B1 >= 6 && B1 <= 14 &&
```

```
C1 >= 5 && C1 <= 14 &&

A2 >= 24 && A2 <= 68 &&

B2 >= 33.5 && B2 <= 90 &&

C2 >= 19.5 && C2 <= 62 &&

A3 >= 24 && A3 <= 70 &&

B3 >= 32 && B3 <= 90 &&

C3 >= 19 && C3 <= 62 &&

A4 >= 28 && A4 <= 70 &&

B4 >= 28 && B4 <= 86 &&

C4 >= 20 && C4 <= 65 &&

AA1 >= 30 && AA1 <= 85 &&

BB1 >= 0.2 && BB1 <= 10 &&

CC1 >= 3 && CC1 <= 15)
                    {

                    std::cout

                    << "D"

                    << std::endl;

                                        }

                                //********* KEEP
VALUE OF CHARACTER OF
E**********\\

                                                if
(B0 >= 0 && B0 <= 50 &&

        C0 >= 5 && C0 <= 55 &&

        A1 >= 3 && A1 <= 35 &&

        B1 >= 25 && B1 <= 90 &&

        C1 >= 15 && C1 <= 85 &&

        A2 >= 3 && A2 <= 40 &&

        B2 >= 25 && B2 <= 90 &&

        C2 >= 15 && C2 <= 80 &&

        A3 >= 0 && A3 <= 40 &&

        B3 >= 25 && B3 <= 75 &&
```

```
                                        //********* KEEP
C3 >= 15 && C3 <= 65 &&                 VALUE OF CHARACTER OF
                                        F**********\\
A4 >= 0 && A4 <= 40 &&                                  if (B0 >=
                                        10 && B0 <= 43 &&
B4 >= 0 && B4 <= 40 &&
                                                C0 >= 12 && C0 <= 72 &&
C4 >= 15 && C4 <= 65 &&
                                                A1 >= 25 && A1 <= 77 &&
AA1 >= 1 && AA1 <= 10 &&
                                                B1 >= 56 && B1 <= 85 &&
BB1 >= 0 && BB1 <= 25 &&
                                                C1 >= 26 && C1 <= 60 &&
CC1 >= 2 && CC1 <= 10 &&
                                                A2 >= 3.3 && A2 <= 65 &&
DD1 >= 35 && DD1 <= 130 )
                        {                       B2 >= 2.3 && B2 <= 64 &&

                                                C2 >= 1.7 && C2 <= 43 &&


                                                A3 >= 1.2 && A3 <= 62 &&

std::cout
                                                B3 >= 1 && B3 <= 64 &&
<< "E"
                                                C3 >= 0.8 && C3 <= 33 &&
<< std::endl;
                                                A4 >= 0.5 && A4 <= 74 &&
                        }
                                                B4 >= 0.5 && B4 <= 66 &&
```

```
C4 >= 1.2 && C4 <= 31 &&

AA1 >= 16 && AA1 <= 50 &&

BB1 >= 3 && BB1 <= 15 &&

CC1 >= 5 && CC1 <= 15)
                        {

std::cout

<< "F"

<< std::endl;

                        }

                //********* KEEP
VALUE OF CHARACTER OF
G**********\\

                        if (B0 >=
0 && B0 <= 10 &&

C0 >= 0 && C0 <= 10 &&

A1 >= 10 && A1 <= 45 &&

B1 >= 5 && B1 <= 30 &&

C1 >= 3 && C1 <= 20 &&

A2 >= 50 && A2 <= 90 &&

B2 >= 20 && B2 <= 90 &&

C2 >= 18 && C2 <= 40 &&

A3 >= 55 && A3 <= 90 &&

B3 >= 40 && B3 <= 90 &&

C3 >= 18 && C3 <= 40 &&

A4 >= 55 && A4 <= 90 &&

B4 >= 54 && B4 <= 90 &&

C4 >= 18 && C4 <= 40 &&

AA1 >= 10 && AA1 <= 50 &&

BB1 >= 1 && BB1 <= 8 &&

CC1 >= 3 && CC1 <= 13 &&

DD1 >= 10 && DD1 <= 50)
                        {

std::cout
```

```
        << "G"

        << std::endl;

                        }

                    //********** KEEP
VALUE OF CHARACTER OF
H************\\

                        if (B0 >=
15 && B0 <= 45 &&

        C0 >= 13 && C0 <= 68 &&

        A1 >= 5 && A1 <= 20 &&

        B1 >= 2 && B1 <= 10 &&

        C1 >= 1.5 && C1 <= 10 &&

        A2 >= 2 && A2 <= 15 &&

        B2 >= 2 && B2 <= 10 &&

        C2 >= 1.5 && C2 <= 10 &&

        A3 >= 60 && A3 <= 85 &&

        B3 >= 26 && B3 <= 90 &&

        C3 >= 12 && C3 <= 50 &&

        A4 >= 60 && A4 <= 85 &&

        B4 >= 26 && B4 <= 86 &&

        C4 >= 12 && C4 <= 50 &&

        AA1 >= 0.5 && AA1 <= 10 &&

        BB1 >= 40 && BB1 <= 85 &&

        CC1 >= 2 && CC1 <= 12 &&

        DD1 >= 20 && DD1 <= 40 &&

        EE1 >= 10 && EE1<= 30 )
                        {

        std::cout

        << "H"

        << std::endl;

                        }

                    //********** KEEP
VALUE OF CHARACTER OF I************\\

                        if (B0 >=
36 && B0 <= 44 &&
```

```
C0 >= 2.7 && C0 <= 44 &&

A1 >= 60 && A1 <= 85 &&

B1 >= 57 && B1 <= 87 &&

C1 >= 24 && C1 <= 40.4 &&

A2 >= 70.9 && A2 <= 87 &&

B2 >= 57 && B2 <= 87 &&

C2 >= 24 && C2 <= 41 &&

A3 >= 70.1 && A3 <= 87 &&

B3 >= 57 && B3 <= 88 &&

C3 >= 24 && C3 <= 42 &&

A4 >= 4 && A4 <= 28 &&

B4 >= 5.4 && B4 <= 11 &&

C4 >= 4.1 && C4 <= 10)
                        {

std::cout
                        << "I"

                        << std::endl;

                                }
                        //********* KEEP
VALUE OF CHARACTER OF
K**********\\
                                if (B0 >=
20 && B0 <= 42 &&

        C0 >= 12 && C0 <= 65 &&

        A1 >= 5 && A1 <= 27 &&

        B1 >= 2 && B1 <= 8 &&

        C1 >= 2 && C1 <= 6 &&

        A2 >= 2 && A2 <= 27 &&

        B2 >= 2 && B2 <= 15 &&

        C2 >= 2 && C2 <= 9 &&

        A3 >= 45 && A3 <= 85 &&

        B3 >= 54 && B3 <= 90 &&

        C3 >= 23 && C3 <= 51 &&
```

```
A4 >= 45 && A4 <= 85 &&

B4 >= 45 && B4 <= 87 &&

C4 >= 24 && C4 <= 50 &&

AA1 >= 10 && AA1 <= 30 &&

BB1 >= 40 && BB1 <= 78 &&

CC1 >= 0.1 && CC1 <= 15 &&

DD1 >= 0 && DD1 <= 40 )
                        {

std::cout

<< "K"

<< std::endl;

                        }

                //********* KEEP
VALUE OF CHARACTER OF
L**********\\

                        if (B0 >=
2 && B0 <= 20 &&

C0 >= 4 && C0 <= 18 &&
```

```
A1 >= 1 && A1 <= 30 &&

B1 >= 4 && B1 <= 10 &&

C1 >= 3 && C1 <= 9 &&

A2 >= 30 && A2 <= 85 &&

B2 >= 29 && B2 <= 89 &&

C2 >= 15 && C2 <= 60 &&

A3 >= 30 && A3 <= 83 &&

B3 >= 29 && B3 <= 89 &&

C3 >= 15 && C3 <= 60 &&

A4 >= 30 && A4 <= 82 &&

B4 >= 30 && B4 <= 89 &&

C4 >= 16 && C4 <= 60 &&

AA1 >= 20 && AA1 <= 75 &&

BB1 >= 2 && BB1 <= 40 &&

CC1 >= 4 && CC1 <= 15 &&
```

```
        DD1 >= 40 && DD1 <= 100 )
                                {


        std::cout


        << "L"


        << std::endl;


                                }

                        //********* KEEP
VALUE OF CHARACTER OF
M***********\\

                                /*if (B0
>=  && B0 <=  &&


        C0 >=  && C0 <=  &&


        A1 >=  && A1 <=  &&


        B1 >=  && B1 <=  &&


        C1 >=  && C1 <=  &&


        A2 >=  && A2 <=  &&


        B2 >=  && B2 <=  &&


        C2 >=  && C2 <=  &&


        A3 >=  && A3 <=  &&


        B3 >=  && B3 <=  &&


        C3 >=  && C3 <=  &&


        A4 >=  && A4 <=  &&


        B4 >=  && B4 <=  &&


        C4 >=  && C4 <=  &&


        AA1 >=  && AA1 <=  &&


        BB1 >=  && BB1 <=  &&


        CC1 >=  && CC1 <=  &&


        DD1 >=  && DD1 <= )
                                {


        std::cout


        << "M"


        << std::endl;


                                }*/
```

```
                    //********* KEEP
VALUE OF CHARACTER OF
N**********\\
                                /*if (B0
>= 10 && B0 <= 50 &&

        C0 >= 0.5 && C0 <= 55 &&

        A1 >= 50 && A1 <= 90 &&

        B1 >= 50 && B1 <= 90 &&

        C1 >= 20 && C1 <= 50 &&

        A2 >= 50 && A2 <= 85 &&

        B2 >= 50 && B2 <= 90 &&

        C2 >= 20 && C2 <= 50 &&

        A3 >= 50 && A3 <= 90 &&

        B3 >= 35 && B3 <= 90 &&

        C3 >= 15 && C3 <= 50 &&

        A4 >= 60 && A4 <= 85 &&

        B4 >= 60 && B4 <= 85 &&

        C4 >= 15 && C4 <= 50 &&

        AA1 >= 0.5 && AA1 <= 15 &&

        BB1 >= 1 && BB1 <= 15 &&

        CC1 >= 5 && CC1 <= 12  &&

        DD1 >= 75 && DD1 <= 170 )
                                {

        std::cout

        << "N"

        << std::endl;

                                }*/

                    //********* KEEP
VALUE OF CHARACTER OF
O**********\\
                                if (B0 >=
5.5 && B0 <= 45 &&

        C0 >= 5.2 && C0 <= 75 &&

        A1 >= 30 && A1 <= 80 &&

        B1 >= 50 && B1 <= 85 &&
```

```
C1 >= 20 && C1 <= 56 &&

A2 >= 28 && A2 <= 80 &&

B2 >= 50 && B2 <= 84 &&

C2 >= 20 && C2 <= 55 &&

A3 >= 21 && A3 <= 80 &&

B3 >= 26 && B3 <= 65 &&

C3 >= 12 && C3 <= 40 &&

A4 >= 23 && A4 <= 80 &&

B4 >= 26 && B4 <= 80 &&

C4 >= 13 && C4 <= 40 &&

AA1 >= 2 && AA1 <= 10 &&

BB1 >= 1 && BB1 <= 14 &&

CC1 >= 4 && CC1 <= 15 &&

DD1 >= 50 && DD1 <= 110 &&

EE1 >= 50 && EE1 <= 130 )
                        {

std::cout

<< "O"

<< std::endl;

                        }

                //********* KEEP
VALUE OF CHARACTER OF
P**********\\

                        if (B0 >=
1 && B0 <= 40 &&

C0 >= 0.97 && C0 <= 27 &&

A1 >= 3.5 && A1 <= 30 &&

B1 >= 4.5 && B1 <= 15 &&

C1 >= 3.5 && C1 <= 10 &&

A2 >= 54 && A2 <= 83 &&

B2 >= 7.5 && B2 <= 23 &&

C2 >= 3.4 && C2 <= 13 &&
```

```
                                                        if (B0 >=
        A3 >= 80 && A3 <= 88 &&                     0 && B0 <= 10 &&

        B3 >= 67 && B3 <= 88 &&                         C0 >= 0 && C0 <= 20 &&

        C3 >= 27 && C3 <= 38 &&                         A1 >= 10 && A1 <= 50 &&

        A4 >= 80.5 && A4 <= 87 &&                       B1 >= 5 && B1 <= 20 &&

        B4 >= 66.6 && B4 <= 87 &&                       C1 >= 3 && C1 <= 15 &&

        C4 >= 27 && C4 <= 38 &&                         A2 >= 50 && A2 <= 90 &&

        AA1 >= 50 && AA1 <= 90 &&                       B2 >= 20 && B2 <= 90 &&

        BB1 >= 3 && BB1 <= 28 &&                        C2 >= 18 && C2 <= 45 &&

        CC1 >= 7 && CC1 <= 13)                          A3 >= 55 && A3 <= 90 &&
                                  {
                                                        B3 >= 40 && B3 <= 88 &&
        std::cout
                                                        C3 >= 18 && C3 <= 45 &&
        << "P"
                                                        A4 >= 55 && A4 <= 90 &&
        << std::endl;
                                                        B4 >= 54 && B4 <= 90 &&
                                  }
                          //********* KEEP
VALUE OF CHARACTER OF                                   C4 >= 18 && C4 <= 44 &&
Q**********\\
                                                        AA1 >= 10 && AA1 <= 85 &&
```

```
BB1 >= 1 && BB1 <= 8 &&

CC1 >= 3 && CC1 <= 13 &&

DD1 >= 10 && DD1 <= 30)
                        {

std::cout

<< "Q"

<< std::endl;

                        }
                //********* KEEP
VALUE OF CHARACTER OF
R**********\\
                        if (B0 >=
20 && B0 <= 45 &&

    C0 >= 5 && C0 <= 40 &&

    A1 >= 15 && A1 <= 35 &&

    B1 >= 4 && B1 <= 20 &&

    C1 >= 3 && C1 <= 15 &&

    A2 >= 7 && A2 <= 30 &&

    B2 >= 3 && B2 <= 20 &&

    C2 >= 2 && C2 <= 15 &&

    A3 >= 70 && A3 <= 85 &&

    B3 >= 30 && B3 <= 80 &&

    C3 >= 10 && C3 <= 40 &&

    A4 >= 70 && A4 <= 90 &&

    B4 >= 70 && B4 <= 90 &&

    C4 >= 10 && C4 <= 40 &&

    AA1 >= 10 && AA1 <= 25 &&

    BB1 >= 50 && BB1 <= 70  &&

    CC1 >= 5 && CC1 <= 15 )
                                {

    std::cout

    << "R"

    << std::endl;

                }
```

```
                    //********* KEEP
VALUE OF CHARACTER OF
S**********\\
                                if (B0 >=
8 && B0 <= 45 &&

        C0 >= 15 && C0 <= 72 &&

        A1 >= 56 && A1 <= 90 &&

        B1 >= 75 && B1 <= 90 &&

        C1 >= 30 && C1 <= 48 &&

        A2 >= 53 && A2 <= 86 &&

        B2 >= 80 && B2 <= 90 &&

        C2 >= 30 && C2 <= 50 &&

        A3 >= 40 && A3 <= 90 &&

        B3 >= 45 && B3 <= 86 &&

        C3 >= 25 && C3 <= 43 &&

        A4 >= 40 && A4 <= 87 &&

        B4 >= 40 && B4 <= 87 &&

        C4 >= 20 && C4 <= 40 &&

        AA1 >= 3 && AA1 <= 10 &&

        BB1 >= 3 && BB1 <= 15 &&

        CC1 >= 4 && CC1 <= 12)
                                {

        std::cout

        << "S"

        << std::endl;

                                }
                    //********* KEEP
VALUE OF CHARACTER OF
T**********\\
                                /*if (B0
>= 5 && B0 <= 40 &&

        C0 >= 0 && C0 <= 60 &&

        A1 >= 30 && A1 <= 75 &&

        B1 >= 40 && B1 <= 90 &&

        C1 >= 20 && C1 <= 60 &&
```

```
A2 >= 30 && A2 <= 80 &&

B2 >= 65 && B2 <= 90 &&

C2 >= 20 && C2 <= 60 &&

A3 >= 45 && A3 <= 90 &&

B3 >= 25 && B3 <= 90 &&

C3 >= 10 && C3 <= 52 &&

A4 >= 40 && A4 <= 90 &&

B4 >= 45 && B4 <= 90 &&

C4 >= 10 && C4 <= 52 &&

AA1 >= 1 && AA1 <= 8 &&

BB1 >= 1 && BB1 <= 30 &&

CC1 >= 3 && CC1 <= 15 &&

DD1 >= 50 && DD1 <= 150)
                        {

std::cout
                        << "T"

                        << std::endl;

                                        }*/

                                //********* KEEP
VALUE OF CHARACTER OF
U**********\\
                                        if (B0 >=
15 && B0 <= 45 &&

        C0 >= 13 && C0 <= 68 &&

        A1 >= 5 && A1 <= 20 &&

        B1 >= 2 && B1 <= 10 &&

        C1 >= 1.5 && C1 <= 10 &&

        A2 >= 2 && A2 <= 15 &&

        B2 >= 2 && B2 <= 10 &&

        C2 >= 1.5 && C2 <= 10 &&

        A3 >= 60 && A3 <= 85 &&

        B3 >= 26 && B3 <= 90 &&
```

```
C3 >= 12 && C3 <= 50 &&

A4 >= 60 && A4 <= 85 &&

B4 >= 26 && B4 <= 86 &&

C4 >= 12 && C4 <= 50 &&

AA1 >= 0.5 && AA1 <= 10 &&

BB1 >= 40 && BB1 <= 85 &&

CC1 >= 2 && CC1 <= 12 &&

DD1 >= 50 && DD1 <= 60  )
                        {

std::cout

<< "u"

<< std::endl;

                        }

                //********* KEEP
VALUE OF CHARACTER OF
V**********\\

                        if (B0 >=
20 && B0 <= 42 &&
```

```
C0 >= 12 && C0 <= 65 &&

A1 >= 5 && A1 <= 27 &&

B1 >= 2 && B1 <= 8 &&

C1 >= 2 && C1 <= 6 &&

A2 >= 2 && A2 <= 27 &&

B2 >= 2 && B2 <= 15 &&

C2 >= 2 && C2 <= 9 &&

A3 >= 45 && A3 <= 85 &&

B3 >= 54 && B3 <= 90 &&

C3 >= 23 && C3 <= 51 &&

A4 >= 45 && A4 <= 85 &&

B4 >= 45 && B4 <= 87 &&

C4 >= 24 && C4 <= 50 &&

AA1 >= 10 && AA1 <= 30 &&

BB1 >= 40 && BB1 <= 78 &&
```

```
        CC1 >= 0.1 && CC1 <= 15 &&

        DD1 >= 60 && DD1 <= 150)
                                    {

        std::cout

        << "V"

        << std::endl;
                                }
                        //********* KEEP
VALUE OF CHARACTER OF
W**********\\
                                if (B0 >=
20 && B0 <= 44 &&

        C0 >= 0.8 && C0 <= 40 &&

        A1 >= 6.9 && A1 <= 30 &&

        B1 >= 4 && B1 <= 15 &&

        C1 >= 3 && C1 <= 12 &&

        A2 >= 10 && A2 <= 35 &&

        B2 >= 3 && B2 <= 15 &&

        C2 >= 2.6 && C2 <= 15 &&

        A3 >= 7.3 && A3 <= 27 &&

        B3 >= 6 && B3 <= 20 &&

        C3 >= 5.5 && C3 <= 17 &&

        A4 >= 39 && A4 <= 78 &&

        B4 >= 45 && B4 <= 72 &&

        C4 >= 20 && C4 <= 50 &&

        AA1 >= 11 && AA1 <= 25 &&

        BB1 >= 5 && BB1 <= 15 &&

        CC1 >= 60 && CC1 <= 95)
                                    {

        std::cout

        << "W"

        << std::endl;

                                }

                //********* KEEP VALUE
        OF CHARACTER OF X**********\\
```

```
                                        if (B0 >=
20 && B0 <= 45 &&

        C0 >= 14 && C0 <= 75 &&

        A1 >= 2 && A1 <= 21 &&

        B1 >= 37.6 && B1 <= 90 &&

        C1 >= 27.7 && C1 <= 70 &&

        A2 >= 53 && A2 <= 86 &&

        B2 >= 20.4 && B2 <= 88 &&

        C2 >= 8.8 && C2 <= 40 &&

        A3 >= 54 && A3 <= 90 &&

        B3 >= 21 && B3 <= 87 &&

        C3 >= 9 && C3 <= 40 &&

        A4 >= 59 && A4 <= 87 &&

        B4 >= 21.8 && B4 <= 87 &&

        C4 >= 9.2 && C4 <= 40 &&

        AA1 >= 70 && AA1 <= 95 &&
        BB1 >= 3 && BB1 <= 15 &&

        CC1 >= 5 && CC1 <= 15)
                                        {

        std::cout

        << "X"

        << std::endl;

                                        }

                                //********* KEEP
VALUE OF CHARACTER OF
Y**********\\

                                        if (B0 >=
0 && B0 <= 20 &&

        C0 >= 1.5 && C0 <= 27 &&

        A1 >= 80 && A1 <= 90 &&

        B1 >= 66 && B1 <= 86.8 &&

        C1 >= 28 && C1 <= 36.5 &&

        A2 >= 76 && A2 <= 86 &&

        B2 >= 66 && B2 <= 85.9 &&
```

```
C2 >= 28 && C2 <= 38 &&

A3 >= 78 && A3 <= 90 &&

B3 >= 66 && B3 <= 86 &&

C3 >= 28 && C3 <= 37 &&

A4 >= 2 && A4 <= 27 &&

B4 >= 6 && B4 <= 13.5 &&

C4 >= 5 && C4 <= 11.4)
                        {

std::cout

<< "Y"

<< std::endl;

                        }

                                if
(stop == 100 )

                        {

        std::cout

                << "Please Type Any Keys Press
 Enter To exit"

                << std::endl;

                std::cin.get();

                                        }

                                }

                        }

                }

}

void
SampleListener::onFocusGained(const
Controller& controller) {

}

void
SampleListener::onFocusLost(const
Controller& controller) {

}

void
SampleListener::onDeviceChange(const
Controller& controller) {

        const DeviceList devices =
controller.devices();

        for (int i = 0; i <
devices.count(); ++i) {

        }
```

```
}

void
SampleListener::onServiceConnect(con
st Controller& controller) {

}

void
SampleListener::onServiceDisconnect(c
onst Controller& controller) {

}

int main(int argc, char** argv) {

        SampleListener listener;

        Controller controller;

        controller.addListener(listener);

        if (argc > 1 && strcmp(argv[1], "-
-bg") == 0)
        controller.setPolicyFlags(Leap::
Controller::POLICY_BACKGROUND_FRA
MES);

        std::cin.get();

        return 0;

}
```